WEST Search History

| Hide Items | Restore | Clear | Cancel |

DATE: Wednesday, March 10, 2004

Hide?	Set Name	Query	Hit Count
		DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR	
☐	L24	L23 and l22	31
☐	L23	plurality adj2 (computer or device)	80817
☐	L22	L21 and ((communication or communicating) adj3 (interface or interfacing))	127
☐	L21	19990126	776
☐	L20	display same (display adj2 (control or controller)) same ((display or displaying) adj3 (interface or interfacing))	1409
☐	L19	19990126	18334
☐	L18	l17 and l9	36851
☐	L17	(display or displaying or console) near8 (manager or management or center or central or monitor or monitoring or control or controller)	417838
☐	L16	L15 and l9	39
☐	L15	l11 and ((input or inputting) near8 (mean or device))	59
☐	L14	L13 or l12	58
☐	L13	L11 and l10	28
☐	L12	L11 and l9	58
☐	L11	l5 and l7	87
☐	L10	((communication or communicating) near5 (interface or interfacing) and (display or displaying) near5 (interface or interfacing))	15735
☐	L9	((display or displaying) near5 (interface or interfacing))	59751
☐	L8	(display near5 (interface or interfacing))	56745
☐	L7	(plurality adj4 computer)	25246
☐	L6	((display or displaying) near8 (position or size))	179187
☐	L5	19990126)	834
☐	L4	(display near8 (controller or control or controlling) and window near8 (data or information) and (monitor or monitoring or console) near8 (display or displaying) near8 computer)	1589
☐	L3	(display near8 (controller or control or controlling))	293592
☐	L2	(window near8 (data or information))	52495
☐	L1	((monitor or monitoring or console) near8 (display or displaying) near8 computer)	23637

END OF SEARCH HISTORY

<u>First Hit</u>

□ ▓ Generate Collection ▓

L16: Entry 1 of 39   File: PGPB   Jun 6, 2002

DOCUMENT-IDENTIFIER: US 20020067318 A1
TITLE: <u>DISPLAY CONTROL</u> SYSTEM AND ITS CONTROL METHOD, SWITCHING DEVICE, CONNECTION
DEVICE, PERIPHERAL DEVICE, PERIPHERAL DEVICE SYSTEM, AND THEIR CONTROL METHOD, AND
COMPUTER READABLE MEMORY

<u>Abstract Paragraph</u>:
The presence/absence of an image information request signal HSYNC output from each
of a plurality of display devices is monitored. Based on the <u>monitoring result,</u>
<u>image information stored in an image memory of a display controller incorporated in</u>
<u>a host computer is distributed to each display</u> device.

<u>Application Filing Date</u>:
19980710

<u>Summary of Invention Paragraph</u>:
[0001] The present invention relates to a <u>display control system, which connects a</u>
<u>display controller having an image memory for storing image information, and a</u>
<u>plurality of display devices, and controls the plurality of display devices to</u>
<u>display an image on the basis of the image information, and its control</u> method.

<u>Summary of Invention Paragraph</u>:
[0002] The present invention also relates to a switching device and connection
device, which connect a plurality of peripheral devices to a single host equipment,
and a peripheral device system using these devices. More specifically, the present
invention relates to a switching device which connects a <u>plurality of printers to a</u>
<u>host computer</u> having one printer interface, and a peripheral device system.
Furthermore, the present invention relates to a branch device suitable for building
a display system that connects a plurality of display devices to a single host
equipment to <u>display, a peripheral device system using that device, and their</u>
<u>control</u> method.
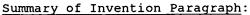
<u>Summary of Invention Paragraph</u>:
[0003] As a display device for a computer equipment or the like, a CRT display
device is generally known. However, since the CRT display device requires a
considerable depth in the thickness direction of its display screen, it has a large
volume as a whole, and the entire system can hardly be made compact. <u>Display</u>
<u>control of such CRT display device must use a CRTC (CRT controller) or the like to</u>
<u>refresh display</u> data, and is complex.

<u>Summary of Invention Paragraph</u>:
[0005] A <u>display controller for such FLCD need not refresh the screen all the time</u>
<u>unlike a CRT display controller</u>. By preferentially updating the display contents of
a display area corresponding to a portion where the contents of a display memory
have been updated, data can be displayed even on a large screen without dropping
the refresh rate.

<u>Summary of Invention Paragraph</u>:
[0007] On the other hand, as a <u>display control system that displays</u> different
contents of image information on a plurality of display devices, the following
three systems are known.

Summary of Invention Paragraph:
[0009] A <u>plurality of host computers</u> are connected via a LAN, and <u>display devices</u> <u>are connected to these host computers via display controllers</u>.

Summary of Invention Paragraph:
[0011] A plurality of <u>display controllers</u> are connected to a single host computer. <u>Display devices are connected to these display controllers</u>.

Summary of Invention Paragraph:
[0013] A <u>display memory on a single host computer and single display controller is</u> <u>logically divided into a plurality of memory areas, which are assigned to a</u> <u>plurality of display</u> devices connected.

Summary of Invention Paragraph:
[0014] However, in order to output identical or different image information contents to a plurality of conventional <u>display devices, display controllers for</u> <u>controlling the display</u> devices are required in correspondence with the number of display devices.

Summary of Invention Paragraph:
[0017] This system requires high cost since one <u>display device requires one each</u> <u>host computer and display controller</u>. Since a <u>plurality of host computers</u> must be controlled, a large-scale, complex control program is required.

Summary of Invention Paragraph:
[0019] This system requires high cost since one <u>display device requires a single</u> <u>display controller</u>. Also, since the number of <u>display controllers that can be</u> <u>connected to a single host computer is limited, the number of display</u> devices that can be connected is inevitably limited.

Summary of Invention Paragraph:
[0029] The present invention has been made in consideration of the above-mentioned problems, and has as its object to provide a flexible <u>display control system, which</u> <u>can control a plurality of display devices by a single display controller and is</u> <u>not restrained by the number of display devices connected, its control</u> method, and a computer readable memory.

Summary of Invention Paragraph:
[0034] It is still another object of the present invention to provide a connection device which automatically determines the peripheral device that communicates the control information, and other peripheral devices in a peripheral device system which connects a plurality of peripheral devices by branching them from a single communication bus and <u>controls the peripheral devices to display identical image</u> <u>data, its control</u> method, and a computer readable memory.

Summary of Invention Paragraph:
[0035] It is still another object of the present invention to provide a peripheral device which automatically determines the peripheral device that communicates the control information, and other peripheral devices in a peripheral device system which connects a plurality of peripheral devices by branching them from a single communication bus and <u>controls the peripheral devices to display different image</u> <u>data, its control</u> method, and a computer readable memory.

Summary of Invention Paragraph:
[0036] In order to achieve the above objects, a <u>display control</u> system according to the present invention comprises the following arrangement.

Summary of Invention Paragraph:
[0037] That is, a <u>display control system which connects a display controller having</u>

an image memory for storing image information to a plurality of display devices,
and controls the plurality of display devices to display an image on the basis of
the image information, comprises:

Summary of Invention Paragraph:
[0039] distribution means for distributing the image information stored in the
image memory of the display controller to each of the plurality of display devices
on the basis of a monitoring result of the monitoring means.

Summary of Invention Paragraph:
[0040] In order to achieve the above objects, a method of controlling a display
control system according to the present invention comprises the following
arrangement.

Summary of Invention Paragraph:
[0041] That is, a method of controlling a display control system which connects a
display controller having an image memory for storing image information to a
plurality of display devices, and controls the plurality of display devices to
display an image on the basis of the image information, comprises:

Summary of Invention Paragraph:
[0043] the distribution step of distributing the image information stored in the
image memory of the display controller to each of the plurality of display devices
on the basis of a monitoring result in the monitoring step.

Summary of Invention Paragraph:
[0044] In order to achieve the above objects, a display control system according to
the present invention comprises the following arrangement.

Summary of Invention Paragraph:
[0045] That is, a display control system which connects a display controller having
an image memory for storing image information to a plurality of display devices,
and controls the plurality of display devices to display an image on the basis of
the image information, comprises:

Summary of Invention Paragraph:
[0047] control means for controlling the display device designated by the
designation means to display the image based on the image information.

Summary of Invention Paragraph:
[0048] In order to achieve the above objects, a method of controlling a display
control system according to the present invention comprises the following
arrangement.

Summary of Invention Paragraph:
[0049] That is, a method of controlling a display control system which connects a
display controller having an image memory for storing image information to a
plurality of display devices, and controls the plurality of display devices to
display an image on the basis of the image information, comprises:

Summary of Invention Paragraph:
[0051] the control step of controlling the display device designated in the
designation step to display the image based on the image information.

Summary of Invention Paragraph:
[0053] That is, a computer readable memory which stores a program code for
controlling a display control system which connects a display controller having an
image memory for storing image information to a plurality of display devices, and
controls the plurality of display devices to display an image on the basis of the
image information, comprises:

□ ▓▓▓ Generate Collection ▓▓▓

L16: Entry 9 of 39   File: USPT   Mar 6, 2001

DOCUMENT-IDENTIFIER: US 6198462 B1
TITLE: Virtual display screen system

Abstract Text (1):
Disclosed is a computerized data display system including a computer operating in
accord with a window display management system for the display and control of a
plurality of data windows on a display screen of a display device. The data windows
are displayed on the display screen in a spatial relation corresponding to the
field of view seen from a preselected viewing location selected by means of a
control signal provided as an input to the computer. A head coupled image display
device, coupled to the display screen of the display device, is adapted to display
the data windows appearing on the display screen of the display device separately
to each eye of a user to create a binocular, stereoscopic virtual screen image to
the user that has a virtual screen size independent of the size of the display
screen of the display device. A user controlled input position sensor, coupled to
the computer, is adapted for generating the position control signal as an input to
the computer to selectively change the selected viewing location.

Application Filing Date (1):
19941014

Brief Summary Text (12):
Generally, the present invention may be embodied in a computerized data display
system which includes a computer operating in accord with a windowing operating
system, such as X-Windows, for the display and control of a plurality of data
windows on at least one display screen of at least one display device. The data
windows are displayed on the display screen in a spatial relation corresponding to
the field of view seen from a preselected viewing location selected by means of a
control signal provided as an input to the computer. A head coupled image display
device, is adapted to display the data windows appearing on each display screen
separately to each eye of a user to create a binocular virtual screen image to the
user that has a virtual screen size independent of the size of the physical display
screen. A sensor that tracks the position of the user's head is coupled to the
computer, and is adapted to selectively change the selected viewing location.

Brief Summary Text (14):
In summary then, the present invention is a novel combination of advanced hardware
and software that enables an entire new paradigm for user interaction with
computers by enlarging the display area to an arbitrary size and providing the
means for new modalities of interaction. It is build upon standard X windows
software and provides interface capability for use with a head mounted display and
head tracker. This capability, as well as the design and construction of novel
methods for interaction and visualization with the virtual screen of the present
invention has been reduced to practice in a prototype system.

Brief Summary Text (16):
The invention is described as a combination of hardware and software that will make
a video display screen appear to be arbitrarily large. One novel aspect of the
invention is the combined use of a window system, physical display hardware in the
form of a head-mounted stereoscopic display (HMD), head tracking, and user

interface control software to produce a virtual screen of apparently large size. This system also allows the user to place windows or other data in any three dimensional position within the working environment of the head tracking and control software. The virtual screen of the present invention is not limited to a two dimensional or perspective view of data, but may also represent three dimensional stereoscopic imagery in the form of left and right eye views to the user.

Brief Summary Text (17):
If a see-through HMD is used instead of a solid surface HMD, the user's standard computer monitor can be used to show the main window, and "offscreen" windows can be arranged at lower resolution around the computer monitor. This see-through variant still allows the user to place windows in an arbitrarily large virtual display, utilizing the standard computer monitor while allowing the user to see his immediate surroundings. The HMD can be lower resolution since it is used more for organization purposes than for reading. Peripheral windows would be moved onto the main computer monitor for close inspection.

Brief Summary Text (21):
2. Increasing productivity through a more intuitive display interface (using head tracking);

Brief Summary Text (23):
4. Simplifying three dimensional visualization by using the same display interface for two or three dimensional visuals;

Detailed Description Text (3):
Generally, the present invention is preferably embodied in a computerized data display system 10 and would include, as shown in FIGS. 1 and 2, a computer or workstation 12 operating under a windowing operating system such as X-Windows, for the display and control of a plurality of data windows 18 on display screens 14 of a display device 16 which may be a head mounted device.

Detailed Description Text (5):
The data windows 18 are displayed on display screens 14 in a spatial relation corresponding to the field of view as seen from a viewing location selected by means of a control signal provided as an input to computer 12 and under the control of the user. The user will then be able to change the viewing point from which the data windows 18 are displayed to correspond with the visual impression derived from the images seen through the head mounted display unit described below.

Detailed Description Text (6):
Workstation 12 preferably contains sufficient memory in either local RAM or other type storage device such as CD-ROM, for retaining the data necessary for the generation and display of data windows 18.
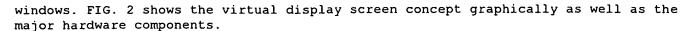
Detailed Description Text (7):
Head mounted display 22 is adapted to display the data windows 18 appearing on display screens 14 of display device 16 separately to each eye of a user to create a binocular, stereoscopic virtual screen image 28 to the user that has a virtual screen size independent of the physical size of display screens 14 of display device 16.

Detailed Description Text (21):
From this information, a new view of the data in the virtual screen memory is constructed and placed in the frame buffer 38. The HMD 22 is fed images from the frame buffer by either of two means: directly from both a left and a right image frame buffer, or by sequentially creating a left, followed by a right image and displaying from a single frame buffer. A 6 degree of freedom input device (one having freedom in six directions of movement) is used to interact with the virtual

windows. FIG. 2 shows the virtual display screen concept graphically as well as the major hardware components.

Detailed Description Text (24):
FIG. 4 shows a block diagram of the software modules comprising the Virtual Screen. Traditional input devices such as the keyboard 42 and mouse 44 are built into the X-server core for rapid update. Input devices such as sensors and Datagloves are interlaced through the X-extensions 40 for X-input and Xvideo. The current prototype interfaces the head tracker directly to the Virtual Screen 46, which then communicates the necessary position information to the underlying X windows code for update of the screen. This approach introduces some latency into the update of the current Virtual Screen view with the benefit of maintaining compatibility with future releases of X windows. The preferred low-latency method interfaces the head tracker directly to the lowest level of the window management system, which for X windows is the X-server.

Detailed Description Text (27):
Look and Drag: A user can view numerous data sets spread out in multiple windows across the Virtual Screen. Typical examples are individuals layers of multi- and hyper-spectral images, demographics, political boundaries, and elevation data. To view a combination of these data sets a user simply centers their FOV on the window of interest, locks it to track the FOV boresight, and drags and stacks it on top of the window containing the data set it is to be combined with. This can be done as often as desired, each time combining the data sets with a chosen operation (typically Boolean).

Detailed Description Text (30):
Pan: Often it is desirable to view spatial data represented by an area larger than the display or window it is being viewed in. This is accomplished by using neutral head movement.

CLAIMS:

1. A computerized data display system comprising:

a computer operating in accord with a window display management system for the display and control of a plurality of data windows on at least one display screen of at least one display device, said plurality of data windows being displayed in a non-overlapping manner on said at least one display screen in a spatial relation corresponding to a field of view seen from a preselected viewing location selected by means of a position control signal provided as an input to said computer;

a head coupled image display device, coupled to said at least one display screen of said at least one display device, for displaying said plurality of data windows appearing on said at least one display screen of said at least one display device separately to each eye of a user to create a binocular virtual screen image to the user that has a virtual screen size independent of the size of said at least one display screen of said at least one display device; and,

user controlled input head position means coupled to said computer, the position means generating said position control signal as an input to said computer to selectively change said selected viewing location.

2. A computerized data display system as in claim 1 further including:

memory means coupled with said computer for retaining a plurality of data to be used by said computer for the generation and display of said plurality of data windows.

8. A computerized data display system as in claim 1 wherein said head coupled image

display device further displays said plurality of <u>data windows</u> separately to each eye of a user to create a binocular stereoscopic virtual screen image to the user.

☐ | Generate Collection |

L16: Entry 12 of 39   File: USPT   Aug 29, 2000

DOCUMENT-IDENTIFIER: US 6112237 A
TITLE: Electronic monitoring system and method for externally monitoring processes
in a computer system

Application Filing Date (1):
19961126

Brief Summary Text (4):
Over the past several years, the number of installed computer systems has
increased. In addition, these computer systems are getting more and more
complicated with many subsystems. Also, computer systems are being integrated with
others across a room, in another building, and perhaps in another part of the
country. This proliferation of computer systems has resulted in an increase in
productivity of certain types of employees, but at the same time it has created a
need for many new employees. In particular, new computer technicians have been
hired to monitor and repair the new computer systems. To help aid in the monitoring
process many of these computer systems include an operating console which is
displayed on a video display. The operating console shows the status of various
system components like disk drives, chillers, data communication lines, etc.

Drawing Description Text (9):
FIG. 8 is a screen display of the Icon Control window.

Detailed Description Text (20):
Each window shown on the display can be divided into a client region, used for
entering and displaying data through graphics or with key input, and a window
function region on the periphery, used to control the window.

Detailed Description Text (29):
To display the Icon Control window, FIG. 8, double click on the Icon Control
button. This window allows the user to put a specific message into the system log.
The buttons on this screen perform the following functions. The Color Box allows
selection of the color of the message being entered into the log. The ICS session
uses different colors to indicate the priority of a message. The Message Box allows
the operator to add messages to the system message log. The Operator name Box
allows the operator to type in a name up to eight characters long. The Log comment
Box allows the operator to add a comment to the system message log. This message is
displayed on the Filtered Message Display, FIG. 6. The comment may be up to 50
characters long. When a special log message has been typed in, save the message in
the log by clicking on the "OK" button. To return to the previous window without
saving any entries, click on the "Cancel" button.

Detailed Description Text (30):
The System Status Summary window, shown in FIG. 9, (called the System Summary
window) displays the status of the data center, hardware, or operating system and
I/O box signal. Information is displayed in specific colors to indicate the status,
such as the room, CPU, and operating system status. To display a window with
detailed information on the specified room, CPU, or operating system, double click
on the appropriate button.

Detailed Description Text (47):
There are six buttons on the H/W Configuration Window, FIG. 19. Each button
displays the operational status. If anything unusual happens, the appropriate
button will change color. The top button displays the status of the CPU (for
example, "Tokyo 9021 normal"). The middle button displays the status of the CPU
channel. The bottom four buttons display the status of the I/O units (DASD, Tape,
Printer and Miscellaneous). These screen relations are in "Normal Extension" mode.
In the other modes, the screen relation will be different. In "Not Extension" mode,
if the top button is clicked, the Emulator window will be displayed. In "Express
Extension" mode, if the top button is clicked, the CPU log window will be
displayed. To change the color or name of a button, click on the desired button.
Click on the Select Icon menu. Then display the Icon Control window. The operator
will be able to change the color or name from this window.

Detailed Description Text (241):
Referring now to FIG. 46, there is shown a method of converting the IOCP to Mark3.
First the message "Start of Converter for IOCP to Mark3" is displayed. The command
line is parsed 1024. A help message is displayed 1026 upon user request. Input and
output files are opened 1028. If there was an error in opening 1030, then an error
message is displayed 1032 and the process continues at 1034. Alternatively, source
statements are extracted 1036 from the input file. If necessary, the CHPID is
converted 1038. Then, if necessary, the control unit is converted 1040. The input
file pointer is closed 1042. The input file device is opened 1044. If there was an
error 1046 in opening, then an error message is displayed 1032 and the process
continues at 1034. Alternatively, source statements are extracted 1048 from the
device input file. The I/O device is converted 1050. The device input file is
closed 1052. The output file is written to 1054. The output file is closed 1056. If
there were any errors in the process, the message "IOCP Convert" is displayed 1058
and the number of errors found is displayed 1058. The module ends 1060.
Alternatively, the message "IOCP Convert Successfully Complete" is displayed 1062
and the module ends 1060.

Detailed Description Text (283):
The monitoring system 100 also includes a graphical user interface 136 for
presenting data as graphical information to a user on a display device 138 and an
input interface 122 for receiving data input by the user on various input devices
like a keyboard 124 and a mouse 126. The processor 110 enables editting of the
processing rules based on user input provided through the input interface 122.

CLAIMS:

5. The monitoring system of claim 4 wherein the plurality of processes are executed
by a computer system in one of several locations selected from a group consisting
of: a single processor, at least two processors, a single computer, at least two
computers, a single network of computers, at least two networks of computers, a
single enterprise, at least two enterprises, a single data center, and at least two
data centers.

6. The monitoring system of claim 4 wherein the plurality of processes are executed
by a computer system and another entity selected from a group consisting of: a
private branch exchange (PBX) system, a telecommunications switch, a security
system, a building environmental control system, a humidity control, a temperature
control, a heating ventilating or air conditioning (HVAC) control, power line, and
a power generator.

13. The monitoring system of claim 1 further comprising graphical user interface
means for presenting data as graphical information to a user on a display device
and input means for receiving data input by the user.

19. The method of claim 17 wherein the receiving step comprises receiving the

preexisting computer system generated signals from a plurality of processes which includes the particular process, the <u>plurality of processes being executed by the computer</u> system in one of several locations selected from a group consisting of: a single processor, at least two processors, a single computer, at least two computers, a single network of computers, at least two networks of computers, a single enterprise, at least two enterprises, a single data center, and at least two data centers.

22. The method of claim 17 further comprising steps of presenting data as graphical information to a user on a display <u>device, receiving data input by the user through an input device, and editing the processing rules based on user input provided through the input device</u>.

☐ ▓▓▓▓ Generate Collection ▓▓▓▓

L16: Entry 13 of 39   File: USPT   Aug 15, 2000

DOCUMENT-IDENTIFIER: US 6104359 A
** See image for <u>Certificate of Correction</u> **
TITLE: Allocating display information

<u>Abstract Text</u> (1):
To provide an expanded computer desktop working area, a forking driver is removably
inserted logically between a graphical device <u>interface program and a plurality of</u>
<u>display device driver programs driving a plurality of computer monitor display</u>
screens. When inserted, the forking driver configures parameters for the screens to
recognize capabilities common to the screens while also preserving significant
capabilities of one of the screens representing a primary screen. The forking
driver intercepts a function call directed to the device driver program
corresponding to the primary screen and processes the function call to cause one or
more of the device driver programs to change one or more screens in a manner
consistent with the expanded working area.

<u>Application Filing Date</u> (1):
<u>19970124</u>

<u>Brief Summary Text</u> (2):
A typical personal <u>computer system includes (FIG. 1) a computer monitor 6 having a</u>
<u>computer screen 8 displaying</u> information to an end-user of the system. Often, the
information is displayed using a presentation "shell" program 10 having a graphical
user interface ("GUI"). The shell program is provided by operating system software
(e.g., Microsoft.RTM. Windows.RTM.95) running on the system. On the monitor's
screen, the GUI provides one or more display windows, each defining a specific sub-
area within the screen's entire display area, or encompassing the entire display
area. The GUI allows the end-user or an application program 12 (e.g., a word-
processing program) to create one or more display windows corresponding to one or
more tasks. Once created, the display windows can be positioned at different parts
of the computer screen by the application program or by the end-user manipulating
an <u>input device</u> such as a computer mouse or a keyboard.

<u>Brief Summary Text</u> (3):
To display <u>information in one or more of the display windows,</u> an application
program running on the system uses a graphical device interface subsystem 14
("GDI") provided by the operating system. GDI serves as a link between the
application program and a graphics device, such as a video display adapter 15
(e.g., a plug-in card) included with the computer to provide graphics output
signals to the computer monitor. More particularly, GDI lies logically between the
application program and a graphics device driver 16, which is a computer program
designed to <u>control a particular graphics hardware device such as the video display</u>
adapter. The device driver is not technically part of the operating system. Rather,
the device driver is a software module (often provided by the vendor of the video
display adapter) supplementing the operating system for updating the status of the
video display adapter and the monitor in response to calls made to functions of the
operating system in general and to functions of GDI in particular.

<u>Brief Summary Text</u> (4):
Communication between GDI and the device driver is facilitated by a device driver

interface 18 ("DDI"). DDI is a set of functions provided by the device driver for access by GDI. Information is passed between GDI and the <u>device driver through input</u> and output parameters of these functions. Some of the functions are necessary for communication between GDI and DDI, while others are optional. GDI calls to an "Enable" function initialize the device driver and cause GDI to receive information about device-driver-supported DDI functions. GDI itself handles any operations not supported by the device driver. Most of the functions are exposed to GDI through an array of pointers residing in a pointer table.

Brief Summary Text (13):
When the operating system is started, the device context of the video display adapter is initialized by another operating system subsystem "USER". USER provides functions relating to the GUI, including functions to create, move, size, and remove screen objects such as display windows, selection menus appearing in the display windows, graphical icons, and the like. For example, an application program can call a USER function to remove a display window associated with the application program. USER also controls other resources such as a sound driver, a timer, and communications ports. In addition, end-user <u>input from a mouse, a keyboard, or other input device</u> is directed by USER to an application program.

Brief Summary Text (15):
After the device context is initialized, USER queries the device for several of its characteristics and capabilities so as to properly maintain the desktop. Initializing the device context provides USER with information about icons, key and mouse cursors, and bitmap resources for defining the appearance of other screen objects such as graphical buttons <u>controlling the presence of a display</u> window. Once initialized by USER, the device context typically remains fairly stable for the duration of the Windows.RTM. session, i.e., until the operating system is restarted for some reason.

Brief Summary Text (30):
The technique may be implemented in hardware or software, or a combination of both. Preferably, the technique is implemented in computer programs executing on programmable computers that each include a processor, a storage medium readable by the processor (including volatile and non-volatile memory and/or storage elements), at least one <u>input device,</u> and at least one output device. Program code is applied to data entered using the <u>input device</u> to perform the method described above and to generate output information. The output information is applied to one or more output devices.

Detailed Description Text (10):
With reference to FIGS. 2-7, the end-user initiates a process of enabling (i.e., attaching) an additional display adapter (e.g., the second adapter 260 mentioned above) by running a subprogram known as "<u>Control Panel--Display</u>"(step 1000). Provided by the Windows.RTM. "shell"subsystem 205, "<u>Control Panel--Display</u>" allows the end-user to select a color bit-depth, a resolution, and a logical position for an additional screen (e.g., the second screen 250 mentioned above) corresponding to the second adapter. The additional screen's logical position is determined relative to the logical position of the first screen 230. All of these selections by the end-user may be changed dynamically later.

Detailed Description Text (11):
"<u>Control Panel--Display</u>" then calls a "ChangeDisplaySettings()" function of the Windows.RTM. USER subsystem 280 (step 1010). In turn, "ChangeDisplaySettings()" calls a GDI function "CreateDC()"for creating an additional device context 310 for the additional screen and returning an additional pointer "hDC"for this context 310 (step 1020). Note that for the first screen there is an already-existing hDC pointing to an already-existing device context 320 created when the operating system was started. Commonly used by applications for performing graphics operations, the device context encapsulates all of the graphics information about

the additional adapter. The graphics information includes attributes such as color bit-depth, size in pixels, resolution in dots per inch or centimeter, current position for insertion of text, current text font selected, current brush selected, and current pen selected.

Detailed Description Text (25):
device driver and adapter combination uses a color bit-depth of 8 and a particular palette, that palette is also used by all attached additional device driver and adapter combinations configured to employ a color bit-depth of 8 and a palette. Such use eases compatibility with older application programs, but is not necessary; other embodiments may allow the use of different palettes. To match the palette information, the DSD fields of the second device block are set to reflect use of the same palette used by the first device driver and adapter. In addition, to ensure synchronization of the first and second device driver and adapter combinations, a palette translate bit and a palette translate table are copied from the first DSD fields to the second DSD fields. The translate bit and the translate table are used in drawing bitmapped images in background display windows using palettes that are different from the palette used by a foreground display window. Because the foreground display window controls the palette used, copying the translate bit and the translate table allows the background to be displayed as accurately as possible. In addition, a patch is made to ensure that the DIB engine behaves properly with respect to multiple device driver and adapter combinations using palettes. To determine whether two individually selected palettes are in fact the same palette, the DIB engine compares pointers to the palettes, instead of comparing the palettes directly. Therefore, for each additional device driver and adapter combination, a corresponding pointer known as "deBitmapInfo" in a data structure LPDIBENGINE is patched to match the deBitmapInfo pointer corresponding to the first device driver and adapter combination.

CLAIMS:

1. A method of allocating display information among a plurality of display screens in a computer system, the method comprising:

identifying a first function provided by a first display device driver and a second function provided by a second display device driver, the functions and the device drivers configured to allow at least one of the display screens to be updated in accordance with the display information, the first function and the first driver corresponding to one of the screens, the second function and the second driver corresponding to another of the screens;

replacing an original pointer to the first function with a redirection pointer;

configuring a capability attribute, the configuration reflecting a capability common to both drivers;
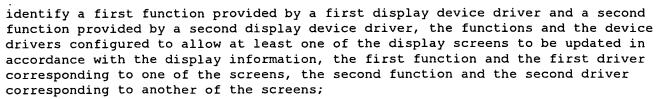
using the redirection pointer to redirect a function call comprising the display information, the function call originally directed to the first function;

processing the display information to derive new display information in accordance with the configured capability attribute; and

updating at least one of the screens by causing a new function call directed to one of the functions, the new function call comprising the new display information.

14. Computer software, residing on a computer-readable storage medium, comprising instructions for use in a computer system to allocate display information among a plurality of display screens in the computer system, the instructions causing the system to:

identify a first function provided by a first display device driver and a second function provided by a second display device driver, the functions and the device drivers configured to allow at least one of the display screens to be updated in accordance with the display information, the first function and the first driver corresponding to one of the screens, the second function and the second driver corresponding to another of the screens;

replace an original pointer to the first function with a redirection pointer;

configure a capability attribute, the configuration reflecting a capability common to both drivers;

use the redirection pointer to redirect a function call comprising the display information, the function call originally directed to the first function;

process the display information to derive new display information in accordance with the configured capability attribute; and

update at least one of the screens by causing a new function call directed to one of the functions, the new function call comprising the new display information.

☐ ▓ Generate Collection ▓

L16: Entry 14 of 39   File: USPT   Jun 13, 2000

DOCUMENT-IDENTIFIER: US 6075531 A
TITLE: Computer system and method of manipulating multiple graphical user interface
components on a computer display with a proximity pointer

Application Filing Date (1):
19971215

Brief Summary Text (6):
Each computer software application executing in a GUI environment is typically
allocated one or more windows to present information to and/or receive input from a
user. Moreover, a number of computer systems provide the ability to multitask--that
is, to execute more than one computer software application at the same time--such
that windows from multiple applications may be displayed on a computer display
simultaneously. As a result, a large number of windows may be displayed on a
computer system at any given time.

Detailed Description Text (8):
Computer system 20, which may be similar to computer systems 12, 14, may include a
processor such as a microprocessor 21; a number of peripheral components such as a
computer display 22; storage devices 23 such as hard, floppy, and/or CD-ROM disk
drives; a printer 24; and various input devices (e.g., a mouse 26 and keyboard 27),
among others. Computer system 20 operates under the control of an operating system,
and executes various computer software applications, programs, objects, modules,
etc. Moreover, various applications, programs, objects, modules, etc. may also
execute on one or more processors in server 16 or other computer systems 12, 14,
e.g., in a distributed computing environment.

Detailed Description Text (9):
Computer display 22 may include any known manner of visually presenting information
to a user. For example, computer display 22 may be a video monitor, e.g., a
cathode-ray tube (CRT), a liquid crystal display (LCD), or a projection display,
among others. In addition, other types of computer displays, including two
dimensional displays that simulate three dimensions (e.g., virtual reality
headsets), as well as three-dimensional displays such as holographic tanks and the
like, may also be used.

Detailed Description Text (10):
User input may also be received from other known user input devices. For example,
control of a pointer on a display may be handled by a trackball, a joystick, a
light pen, a touch sensitive pad or display, a digitizing tablet, and a keyboard,
among others. In addition, many of such devices include one or more user controls
such as buttons, thumb wheels, sliders and the like. Moreover, voice and/or image
recognition may be used to permit a user to provide voice commands and/or gestures
to provide user input to a computer system. Other user interface devices may also
be used in the alternative.

Detailed Description Text (15):
Computer system 10 is typically under the control of an operating system 30, a
component of which is an event manager, which is illustrated at 32 and is shown as
resident in memory 28. Event manager 32 generally provides in conjunction with a

display manager (not shown) an event-driven graphical user <u>interface (GUI)</u>
<u>environment for handling the display</u> of information to, and the receipt of input
from, a user. However, it should be appreciated that routines consistent with the
invention may also find use in other processes than an event manager. For example,
routines consistent with the invention may find use in various computer software
applications that execute on top of an operating system.

<u>Detailed Description Text</u> (32):
Once the representation of the pointer has been set, a find affected window
boundary segments routine 120 is called to determine which windows, and
specifically, which boundaries segments thereof, are at least partially disposed
within the current proximity range for the pointer. As will be discussed in greater
detail below, routine 120 returns a "gripped list" that indicates each of the
windows and segments thereof that fall within the proximity range of the pointer.
This <u>information is utilized in block 106 to highlight the affected windows</u> so that
a user can discern the windows that would be affected should a grip event be
initiated by the user. Highlighting of affected windows is generally performed by
setting a flag in each window object such that, upon their display, a unique
representation is generated for the windows to distinguish such windows from non-
affected windows.

<u>Detailed Description Text</u> (34):
After the affected windows have been highlighted, <u>control passes to block 108 to</u>
<u>update the display</u>. This has the result of refreshing the pointer to its move
pointer representation, and of refreshing each affected window to display the
affected status thereof. It should be appreciated that in general, updating a
display, or otherwise refreshing components rendered on a display, is well known in
the art.

<u>Detailed Description Text</u> (48):
In either event, control is passed to block 132 after the radius of the grip span
is set. Block 132 clears a gripped list that is maintained by the computer system
to store <u>information pertaining to all of the affected windows</u> for the current
proximity range defined at the current position of the pointer. Any number of data
structures may be utilized to implement the gripped list, e.g., a list of records,
with each record related to one affected window.

<u>Detailed Description Text</u> (50):
Next, decision block 134 is executed, and if another window in the <u>display is found</u>
<u>to be at least partially disposed within the grip span, control</u> is passed to block
136 to retrieve the dimensions of the window. Then, in block 138, the affected
boundary segments (i.e., the intersected boundary segments) in that window. It
should be appreciated that blocks 136, 138 may not be separately required as the
calculation of such information may inherently be performed in some implementations
of block 133.

<u>Detailed Description Text</u> (51):
Based upon the above <u>information, a record in the gripped list is created for the</u>
<u>window</u> in block 139. Within this record, the affected boundary segments are stored.
Control is then returned to block 133 to identify the next window within the grip
span.

<u>Detailed Description Text</u> (58):
Next, block 166, diverts control to one of two program flows depending upon whether
the grip flag has been set. If the grip mode has not been set, block 166 passes
control to block 167 to determine whether the current pointer mode is the normal
mode. If it is not (indicating that the pointer mode is either the move or resize
mode), block 167 calls routine 120 (FIG. 6) prior to passing control to blocks 168
and 170, which respectively highlight the affected windows indicated by routine
120, and update the display. If the current pointer mode is the normal mode, block

167 diverts <u>control to block 170 to update the display</u> without highlighting any windows thereon. In either event, after execution of block 170, routine 160 is terminated. The program flow of routine 160 when the grip flag is not set and either of the move or resize modes are selected generally functions to update the gripped list and the highlighting of the affected windows as the pointer is moved around the display.

<u>Detailed Description Text</u> (61):
Once every window in the gripped list has been processed, block 172 passes <u>control to block 170 to update the display</u> and thereby refresh the position and/or size of the affected windows. Routine 160 then terminates.

<u>Detailed Description Text</u> (88):
As shown in FIG. 18, routine 300 begins at block 302 by determining whether the grip flag has been set. If not, routine 300 is terminated, and the change depth event is ignored. In the alternative, it may be desirable to permit manipulation of the depth of a window prior to setting of the grip flag. If the grip flag has been set, block 302 passes control to block 304 to initiate a loop that processes each window in the gripped list. If no windows are stored in the gripped list, <u>control is immediately passed to block 306 to update the display</u> and terminate the routine. As long as additional unprocessed windows exist in the gripped list, however, control is passed to block 308 to retrieve the next window from the gripped list, then block 310 is executed to determine whether the change depth event is a request to move the affected windows up or down in the window hierarchy, also referred to as "promoting" or "demoting" the windows.

<u>Detailed Description Text</u> (92):
Once all windows in the gripped list have been processed, <u>control is passed by block 304 to block 306 to update the display,</u> and thereafter terminate the routine.

CLAIMS:

1. A method of manipulating a <u>plurality of windows on a computer</u> display, the method comprising:

(a) displaying a <u>plurality of windows on a computer</u> display;

(b) displaying a pointer on the computer display;

(c) defining as affected windows first and second windows from the plurality of windows, each affected window at least partially disposed within a proximity range defined around an origin located proximate the pointer; and

(d) in response to user input to move the pointer, concurrently manipulating the affected windows.

28. A computer system, comprising:

(a) a computer display upon which is displayed a plurality of windows and a pointer;

(b) a user <u>input device configured to receive user input;</u> and

(c) a processor, coupled to the computer display and the user <u>input device,</u> the processor configured to define as affected windows first and second windows from the plurality of windows, each affected window at least partially disposed within a proximity range defined around an origin located proximate the pointer, and, in response to user input to move the pointer, to concurrently manipulate the affected windows.

29. A program product, comprising:

(a) a program configured to perform a method of manipulating a <u>plurality of windows on a computer</u> display, the method comprising:

(1) displaying a <u>plurality of windows on a computer</u> display;

(2) displaying a pointer on the computer display;

(3) defining as affected windows first and second windows from the plurality of windows, each affected window at least partially disposed within a proximity range defined around an origin located proximate the pointer; and

(4) in response to user input to move the pointer, concurrently manipulating the affected windows; and

(b) a signal bearing media bearing the program.

32. A method of manipulating a plurality of graphical user <u>interface components on a computer display,</u> the method comprising:

(a) <u>displaying a plurality of graphical user interface components on a computer display;</u>

(b) displaying a pointer on the computer display;

(c) defining as affected graphical user interface components first and second graphical user interface components from the plurality of graphical user interface components, each affected graphical user interface component at least partially disposed within a proximity range defined around an origin located proximate the pointer; and

(d) in response to user input to move the pointer, concurrently manipulating the affected graphical user interface components.

34. A method of manipulating a <u>plurality of windows on a computer</u> display, the method comprising:

(a) displaying a <u>plurality of windows on a computer</u> display;

(b) displaying a pointer on the computer display;

(c) defining as affected windows first and second windows from the plurality of windows, each affected window at least partially disposed within a proximity range defined around an origin located proximate the pointer; and

(d) in response to user input, concurrently moving at least a portion of each affected window in one of an inwardly and an outwardly direction relative to the origin.

42. A computer system, comprising:

(a) a computer display upon which is displayed a plurality of windows and a pointer;

(b) a user <u>input device configured to receive user input;</u> and

(c) a processor, coupled to the computer display and the user <u>input device,</u> the processor configured to define as affected windows first and second windows from

the plurality of windows, each affected window at least partially disposed within a proximity range defined around an origin located proximate the pointer; and, in response to user input, to concurrently move at least a portion of each affected window in one of an inwardly and an outwardly direction relative to the origin.

43. A program product, comprising:

(a) a program configured to perform a method of manipulating a <u>plurality of windows on a computer</u> display, the method comprising:

(1) displaying a <u>plurality of windows on a computer</u> display;

(2) displaying a pointer on the computer display;

(3) defining as affected windows first and second windows from the plurality of windows, each affected window at least partially disposed within a proximity range defined around an origin located proximate the pointer; and

(4) in response to user input, concurrently moving at least a portion of each affected window in one of an inwardly and an outwardly direction relative to the origin; and

(b) a signal bearing media bearing the program.

46. A method of manipulating a plurality of graphical user <u>interface components on a computer display,</u> the method comprising:

(a) <u>displaying a plurality of graphical user interface components on a computer display;</u>

(b) displaying a pointer on the computer display;

(c) defining as affected graphical user interface components first and second graphical user interface components from the plurality of graphical user interface components, each affected graphical user interface component at least partially disposed within a proximity range defined around an origin located proximate the pointer; and

(d) in response to user input, concurrently moving at least a portion of each affected graphical user interface component in one of an inwardly and an outwardly direction relative to the origin.

☐ Generate Collection

L16: Entry 16 of 39   File: USPT   May 9, 2000

DOCUMENT-IDENTIFIER: US 6061064 A
TITLE: System and method for providing and using a computer user interface with a view space having discrete portions

Abstract Text (1):
A system and method associate each of a plurality of computer applications with a corresponding physical location external to the computer and display a given one of the applications when the user focuses attention on the physical location associated with that application. Preferably the display as a view window in a graphical user interface, and the user has means for moving that window relative to the given application. The computer can be a portable and display device can be head mounted. Preferably an input device enables the user to interact with the given application, and preferably the physical locations bring to mind their associated applications. In some embodiments, an identifier, such as a bar code or a coded transmitter, is placed near each of the physical locations to help detect when the user focuses attention on that particular location. The invention also provides a head mounted unit which projects a visual image to the user wearing it. The unit also includes an object

Application Filing Date (1):
19960827

Brief Summary Text (7):
The most common approach has been window based software. Windowing systems attempt to maximize the use of the screen space of a display terminal by providing overlapping windows and icons. The window operating environment, although useful, is often frustrating to operate. The user is required to spend an inordinate amount of time moving, resizing, and opening and closing various windows and icons on the display space. The opening and closing of a window is often slow. Overlapping windows can be aggravating to the eye. It is also difficult to manipulate information within windows. The physical size of the display terminal limits the size of each window, the number of windows that can be displayed at a given time, and, in the case of graphic intensive applications, is often too small to display an image of an object in its entirety.
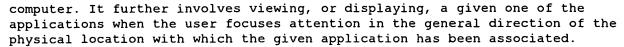
Brief Summary Text (8):
Another approach to increasing the display surface area of a computer is to simply use a larger monitor. Several companies are marketing twenty-eight (28) inch diagonal monitors. These extra-large monitors do increase the display capabilities of the computer to some degree, but the problems outlined above are still present. These monitors are also prohibitively expensive to build and difficult to ship to customers. One such monitor currently on the market weighs over two hundred pounds and is more than thirty inches deep. This monitor is clearly impractical for standard desktop computers.

Brief Summary Text (17):
One aspect of the present invention involves a system and method to enable a user to interact between the real world and a virtual world created by a computer. It involves associating each of a plurality of computer applications with a corresponding physical location external to, and not physically connected to, the

computer. It further involves viewing, or displaying, a given one of the applications when the user focuses attention in the general direction of the physical location with which the given application has been associated.

Brief Summary Text (20):
It is preferred that the user be provided an input device to enable him or her to interact with the given application, such as a mouse, a device which responds to eye movement, a keyboard, a microphone with voice recognition software, a computer network input port, or a radio signal port.

Brief Summary Text (27):
Preferably the unit includes a see through window to enable a user to see his or her physical surroundings when wearing it. Preferably that unit include an additional input device to enable a user wearing it to selectively generate computer readable signals in addition to those generated by the object detector. It is also preferred that the object detector point in the same general direction as the face of a user wearing the unit.

Brief Summary Text (28):
According to another aspect of the invention a computational device is provided which comprises a user interface display, a display generator, and a motion sensor. The display generator generates a display object on the user interface display. The motion sensor is coupled to the display generator, and it directs the display object to move within the user interface display in response to motions sensed by the motion sensor. In a preferred embodiment, the computational device is a hand held computer and the display object is a cursor. In this embodiment, the cursor moves within the user interface display in response to hand motions of the user as detected by the motion sensor.

Drawing Description Text (17):
FIG. 15 illustrates the computing apparatus, with input and output devices, used in certain embodiments of the present invention.

Detailed Description Text (12):
To generate an image of the view port 34 in the virtual view space 40, computer 14 retrieves the pixel information from the frame buffer 42 that corresponds to the view port 34. The pixel information is stored in (1120.times.900) memory locations. The pixel information is subsequently transferred to the head mounted display 22. The pixel information displayed in the head mounted display 22 is referred to as view window 36. The view window 36 includes (1120.times.900) pixels and has the dimensions of (25.degree..times.20.degree.) within the virtual view space 40.

Detailed Description Text (19):
The graphics processor 54, memory 56, arbiter 58, scan line generator 60, graphics output controller 62, FIFO 64 are all provided on a single buffer display card 70. The display timing generator 66 is an electronic card marketed by Reflection Technology along with the head mounted display 22. A ribbon cable (not shown) is used to couple the display buffer card 70 and the display timing generator card 66. The two cards are inserted in the housing of computer 14. The position sensor 24 is wired to the buffer display card 70 through the RS-232 port (not shown) of the computer 14.

Detailed Description Text (70):
The head mounted unit 161 also includes other electronic component necessary to interface to or drive the display 162, the photo detectors 135, the microphone 166, the speaker 168, and the bar code reader 170, as will be understood by those skilled in the art of electronic design. It also includes wires 172 which communicate signals from the head mounted unit to a computer. Preferably this computer is small enough to be worn by the user, such as in a back pack, so the user is free to walk around while using the computer. Such a computer can be

constructed from a SPARCstation computer, of the type described above, with the standard AC power supply replaced with a battery power supply. A computer with enough power to run Sun OS and the Open Windows graphical user interface could be built to fit within a user's pocket using today's technology. Within the coming decade such computers are expected to be small enough to fit within a head mounted unit such as that shown in FIG. 9.

Detailed Description Text (78):
Step 154 performs computation on a communication link associated with an application when user input or computation in steps 153 or 155 requires it. In computers with a graphical user interface the communication links are usually windows created when an application is opened. Such windows communicate from their application to a user by providing visual information or other output. They communicate from the user to their application in response to user mouse clicks, mouse drags, or typing on objects shown in the window. In character based computers the communication links can be screens or non-graphic windows. In audio computers the communication links can be connections to audio input and output.

Detailed Description Text (90):
The bar code 184F is unusual in that it is worn as part of an identity badge on the clothing of a person 204, rather than being stuck on an inanimate object. The windows 180F placed in its associated desktop 182F are those which a user wishes to use when he or she sees the particular person wearing the bar code pattern 184F. One such window might include information about the person wearing the bar code, such as data out of a corporate personnel directory. Another might include matters which the user wishes to discussed with the person, and so on. Other desktops, associated with bar codes worn by other persons, can contain windows associated with such other persons.

Detailed Description Text (95):
FIG. 15 is schematic representation of computing apparatus 230 which enables a user to associate desktops with locations, select desktops by motion relative to such location, and to manipulate visual objects in those desktops. This apparatus preferably includes a portable general purpose computer 14A and a plurality of input and output devices.
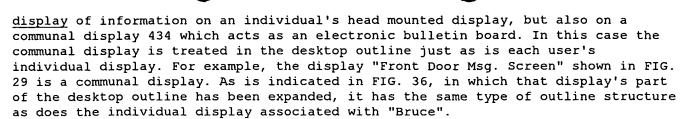
Detailed Description Text (97):
In an embodiment using the head mounted unit 161A, the computing apparatus 230 includes the visual display 162, microphone 166, and speaker, or earphone, 168 of that head mounted unit, and all the standard interface hardware and software necessary to interface to such input/output devices. The apparatus 230 preferably includes connections for a keyboard 232 and a mouse 234. But, as is explained below, such input devices are often not necessary in preferred embodiments of the invention. The apparatus includes the photo detectors 135 for detecting when each of a user's eyes are opened or closed. It also includes a compass detector 236 located on the head mounted unit 161A. This detector includes a magnetic sensor which detects changes in yaw of the head mounted unit relative to the earth's magnetic field lines. The apparatus includes a gravity driven tilt indicator 238 to detect changes in pitch of the head mounted unit. It further includes the bar code detector 170 for detecting when the head mounted unit is pointed at a location labeled with a given bar code pattern.

Detailed Description Text (137):
Steps 362, 364, 366, 368, 370, 372, 373, and 380 list some of the steps that will cause a particular type of action to be performed in response to a particular type of user input from a pointing device, such as a computer mouse or input simulating a mouse in response to head and eye movements as described in FIG. 16.

Detailed Description Text (167):
FIGS. 35 and 36 illustrate how the invention can be used not only to control the

display of information on an individual's head mounted display, but also on a
communal display 434 which acts as an electronic bulletin board. In this case the
communal display is treated in the desktop outline just as is each user's
individual display. For example, the display "Front Door Msg. Screen" shown in FIG.
29 is a communal display. As is indicated in FIG. 36, in which that display's part
of the desktop outline has been expanded, it has the same type of outline structure
as does the individual display associated with "Bruce".

Detailed Description Text (175):
The command interpreter for selecting desktops and generating pointing device
inputs for such a computer is similar to that described above with regard to FIG.
16, except that it uses mouse button input instead of eye blinks to generate mouse
button down signals. When a separate button 466 is pressed, motion of the computer
is not used to move the cursor, allowing extended cursor motion in any given
direction without requiring a corresponding net motion in that direction, the same
way that the closing of both eyes does with the program of FIG. 16.

Detailed Description Text (187):
If the program is in the command recognition mode, the mode in which it responds to
user gestures to control desktop display and cursor functions, the test of step 494
will be met, causing steps 496 and 500 to be performed.

Detailed Description Text (209):
In alternate embodiments of the aspect of the invention, shown with regard to FIG.
11 and the figures that follow it, items other than desktops are associated with
external locations. For example, in some such embodiments windows are associated
directly with external locations, rather than being placed at desired locations in
desktops associated with such locations. In other embodiments of this aspect of the
invention computational objects other than GUI windows are associated with external
locations, including character based windows, images, text messages, audio
messages, OOP type objects, changes in input or output direction, different
locations within a document, changes in modes of user interface operation, the
turning of a display on or off, the zooming of a display's field of vision in or
out, inputs in a command grammar, etc.

Detailed Description Text (211):
In different embodiments of the invention different subsets or superset of the
input and output devices shown in FIG. 15 can be used. For example, in some
embodiments of the invention an individual user's system is not networked, as
described above with regard to FIG. 15, and, thus, has no means for placing windows
in, or seeing what windows are located in, desktops associated with other users. In
other embodiments, the user only carries a display or a smart terminal and much of
the computation which drives his or her display is performed in a stationery
computer, either one associated with the individual user alone, or a time sharing
machine. The connection to such a stationary computer can be either by wire,
optical, or radio link.

Detailed Description Text (213):
The command interpreter shown in FIG. 16 used to control a head-mounted, see-
through display could use input other than eye closings to control its operations.
For example, in other embodiments eye ball direction, speech, head nods, or other
body gestures could be used. Similarly the command gestures used with embodiments
of the invention like that of FIGS. 38 and 39 which views the motion of body parts
can differ. For example, in a system monitoring head position with a video camera,
blinks or facial expressions can be used to simulate mouse clicks or indicate when
pointing is to be interpreted as a command.

Other Reference Publication (5):
Deering, Michael, "Explorations of Display Interfaces for Virtual Reality", Virtual
Reality, 1993 Symposium, pp. 141-147.

CLAIMS:

5. A computing apparatus as in claim 3, further comprising an additional input device to enable a user wearing the head mounted display to selectively generate computer readable signals in addition to those generated by said directional detector.

17. The system of claim 13, further comprising an input device to enable the user to interact with the graphical user interface windows shown in the visual display.

24. The method of claim 20, further comprising the step of enabling the user to interact with the graphical user interface windows shown in the visual display through an input device.

25. The method of claim 24, wherein the step of enabling the user to interact with windows includes enabling the user to interact with such windows using at least one of the following input devices: a mouse, a device for detecting eye movement of the user, a keyboard, a microphone and speech recognition software; a computer network input port, and a radio signal port.

☐ ▓▓▓ Generate Collection ▓▓▓

L16: Entry 19 of 39   File: USPT   Oct 26, 1999

DOCUMENT-IDENTIFIER: US 5973702 A
** See image for Certificate of Correction **
TITLE: Oriented view system having a common window manager for defining application window areas in a screen buffer and application specific view objects for writing into the screen buffer

Abstract Text (1):
An object-oriented view system controls the display of screen graphics for a plurality of application programs, each of which generates graphical information for display in a window assigned to it. The view system has a system window manager which is common to all of the application programs and which defines application window areas on the display screen and corresponding application window storage areas in the display screen buffer. Each application program instantiates a view system object from class information in the computer operating system. Each view system object includes a view object with program code that directly stores screen display information generated by the application into the screen buffer. This arrangement allows the application programs to avoid the conventional "bottleneck" that develops when all of the screen display information must be stored in the screen buffer by the common system window manager.

Application Filing Date (1):
19950607

Brief Summary Text (6):
This invention generally relates to improvements in computer systems and, more particularly, to operating system software for managing drawing areas, called views, inside of a window display area in a graphic user interface.

Brief Summary Text (10):
Each window region generally displays information which is generated by the associated application program and there may be several window regions simultaneously present on the desktop, each representing information generated by a different application program. An application program presents information to the user through each window by drawing or "painting" images, graphics or text within the window region. The user, in turn, communicates with the application by "pointing at" objects in the window region with a cursor which is controlled by a pointing device and manipulating or moving the objects and also by typing information into the keyboard. The window regions may also be moved around on the display screen and changed in size and appearance so that the user can arrange the desktop in a convenient manner.

Brief Summary Text (29):
The lowest or computer hardware level includes the basic computer and associated input and output devices including display monitors, keyboards, pointing devices, such as mice or trackballs, and other standard components, including printers and disc drives. The next or "component driver software" level consists of device-dependent software that generates the commands and signals necessary to operate the various hardware components. The resource control and communication layer interfaces with the component drivers and includes software routines which allocate resources, communicate between applications and multiplex communications generated

by the higher layers to the underlying layers. The view system handles the user interface to basic drawing operations, such as moving and resizing views, activating or inactivating views and redrawing and repainting views. The final user interface layer provides high level facilities that implement the various controls (buttons, sliders, boxes and other controls) that application programs use to develop a complete user interface.

Brief Summary Text (31):
Accordingly, it is an object of the present invention to provide a view system which can interface with application threads in such a manner that the screen display generated by each application thread can be quickly and effectively redrawn.

Brief Summary Text (34):
It is yet another object of the present invention to provide a view system which allows application developers who need detailed control over the screen display process to achieve this control by means of a full set of display control commands which are available, but need not be used by each application thread.

Brief Summary Text (35):
It is yet another object of the present invention to provide a view system which provides application developers with a powerful and flexible drawing environment which includes a virtual coordinate space, arbitrarily shaped views (and windows) and up-to-date drawing state information to facilitate rapid, accurate drawing from multiple threads of execution.

Drawing Description Text (6):
FIG. 4 is a schematic block diagram of a modified computer system showing the interaction between a plurality of application threads, the viewing framework, the window manager, and the screen buffer in order to display graphic information on the display monitor.

Detailed Description Text (3):
Specifically, computer 300 shown in FIG. 3 includes a random access memory (RAM) 306 for temporary storage of information, a read only memory (ROM) 304 for permanent storage of the computer's configuration and basic operating commands and an input/output (I/O) adapter 310 for connecting peripheral devices such as a disk unit 313 and printer 314 to the bus 308, via cables 315 and 312, respectively. A user interface adapter 316 is also provided for connecting input devices, such as a keyboard 320, and other known interface devices including mice, speakers and microphones to the bus 308. Visual output is provided by a display adapter 318 which connects the bus 308 to a display device 322, such as a video monitor. The workstation has resident thereon and is controlled and coordinated by operating system software such as the Apple System/7, operating system.

Detailed Description Text (14):
In the same way that an application framework provides the developer with prefab functionality for an application thread, a system framework, such as that included in a preferred embodiment, can provide a prefab functionality for system level services which developers can modify or override to create customized solutions, thereby avoiding the awkward procedural calls necessary with the prior art application frameworks programs. For example, consider a display framework which could provide the foundation for creating, deleting and manipulating windows to display information generated by an application thread. An application software developer who needed these capabilities would ordinarily have to write specific routines to provide them. To do this with a framework, the developer only needs to supply the characteristics and behavior of the finished display, while the framework provides the actual routines which perform the tasks.

Detailed Description Text (19):

The interaction of an application thread with the view system is illustrated in
more detail in schematic diagram FIG. 5. As previously mentioned, the view system
(illustrated as box 510 in FIG. 5) is an object-oriented program. Accordingly, an
application thread 508 interfaces with the view system by creating and manipulating
"objects". In particular, each application thread creates a view hierarchy object,
for example, view hierarchy object 512 in order to communicate with view system
510. The application thread 508 then communicates with the view hierarchy object
512 by creating a view object 506 and installing it in the hierarchy as shown
schematically by arrow 502. The view system itself is a collection of objects which
is created when the application program is started. The view system 510 interfaces
with the operating system 500 via a data stream 504 to perform window operations on
behalf of the application program and view system 510.

Detailed Description Text (20):
As will hereinafter be described in more detail, each view object 506 includes a
small data store or "cache" area, called the drawing state 514 which is used to
store the associated view visible area and other drawing-related state (coordinate
system etc.). When the application thread desires to redraw the information in one
of its associated views, the view object first checks cache status. If the
information stored in the cache has not been changed or invalidated, then this
information is used to redraw the window. The use of the cache area reduces the
time necessary to complete a redrawing operation.

Detailed Description Text (21):
Since many view objects may be created simultaneously in order to simultaneously
display many views within a window, each view object 506 communicates with the view
system 510 by means of multitask-safe method calls 502. The view system
communicates with the operating system via data stream 504 by creating "stream"
objects which contain the software commands necessary to transfer information from
one object to another. For example, when operating system 500 desires to transfer
information to view system object 510, operating system 500 creates a stream object
which "streams" the data into view system object 510. Similarly, when view system
object 510 desires to transfer information back to operating system 500, view
system object 510 creates a stream object which "streams" the data into window
object 500. Such stream objects are conventional in nature and not described in
detail herein. The stream objects which carry data from operating system 500 to
view system object 510 and the stream objects which carry information from view
system object 510 to operating system 500 are illustrated collectively as arrow
504.

Detailed Description Text (28):
Many application threads further sub-divide the content view into a number of child
views which are independently controlled. These typically include a document view
622, a "toolbar" or "palette" view 616, and, in some cases, a status line view (not
shown). The document view 622 may be equipped with horizontal and vertical scroll
bar views, 618 and 614, that allow objects in the document view to be moved on the
screen. The document view 622 may be further sub-divided into child views 602, 610
and 620 which may also overlap each other (and need not be rectangular). At any
given time usually only one of the child views 602, 610 and 620 is active and only
one view has input "focus". Only the view which has input focus responds to input
actions and commands from the input devices such as the mouse and the keyboard.

Detailed Description Text (30):
The displayed controls are generally selected by means of a mouse or other input
device. The mouse controls a cursor that is drawn on the screen by the operating
system. When the cursor is positioned over the graphic image to be selected, a
button is activated on the mouse causing the view system to respond.

Detailed Description Text (48):
FIG. 19 is a flow chart of an illustrative routine used by the view system to

interact with the input system and distribute a positional event to its destination view object. To simply this flow chart, no special distribution rules are provided or used. The routine starts at step 1900. In step 1902, a positional event is handed to the view system from the <u>input system which received the event from some device</u> that generates positional events, for instance a standard mouse device. The view monitor lock is acquired in step 1904. The view system then makes use of the view hierarchy and a set of extensible rule objects to determine in which view the point is contained. The view system checks to make sure the point is within the "first" (root) view, step 1906. The operating system would not have passed the positional event to this window if the point was not within the window's bounds, so for the root view case, the test should always return true. (If for some reason it returns false, then there may be a "bug" in some portion of the operating system.) Once it is determined that the positional event is within the root view, the view system determines if the point is contained within a child view of the root view, step 1910. If the point is not within a child view, then the "target" for the positional event is the current view. If the point is contained within a child view, then the view system determines which child view, step 1910 and 1914 and continues recursively until the correct, front-most view which contains the positional event has been found (back to 1906 etc.) Once the target has been found, the view monitor is released, step 1916, and the computed target is returned to the input system, step 1918 which will handle all further processing of the event. The routine is completed in step 1920.

CLAIMS:

1. An apparatus for <u>controlling a display</u> device to generate a display having a plurality of windows displayed on a desktop background, each of the plurality of <u>windows being assigned to, and displaying screen information</u> generated by, one of a plurality of application programs, the apparatus comprising:

a screen buffer for storing screen information generated by the plurality of application programs and displaying stored information on the display device;

a view system monitor responsive to screen information generated by the plurality of application programs for dividing the screen buffer into a plurality of storage areas, each of the plurality of storage areas storing the screen <u>information for one of the plurality of windows;</u> and

a plurality of view system objects, each of the plurality of view system objects being part of one of the plurality of application programs and operating to store screen <u>information directly in one of the screen buffer storage areas associated with a window</u> assigned to the one application program.

4. An apparatus according to claim 1 wherein the apparatus comprises an object-oriented operating system for <u>controlling the display</u> device and each of the plurality of view system objects is created by one of the plurality of application programs using class information located in the operating system.

7. An apparatus according to claim 1 wherein each of the view system objects comprises:

a plurality of view objects, each of the plurality of view objects comprising commands for storing screen <u>information into the screen buffer for one window</u> associated with the view system object; and

means controlled by one of the plurality of application programs for arranging the plurality of view objects into a view hierarchy.

9. An apparatus according to claim 1 wherein the apparatus comprises an object-oriented operating system for <u>controlling the display</u> device and the view system

monitor is located in the object-oriented operating system.

10. A method for controlling a computer system having a display device to generate a display having a plurality of windows displayed on a desktop background, each of the plurality of windows being assigned to, and displaying screen information generated by, one of a plurality of application programs, the method comprising the steps of:

A. storing screen information generated by the plurality of application programs in a screen buffer;

B. displaying stored information on the display device;

C. dividing the screen buffer into a plurality of storage areas, each of the plurality of storage areas storing the screen information for one of the plurality of windows using a common view system monitor;

D. creating a plurality of view system objects, each of the plurality of view system objects being part of one of the plurality of application programs; and

E. each of the plurality of view system objects storing screen information directly in a storage area associated with a window assigned to the application program of which the view system object is a part.

12. A method according to claim 11 wherein the computer system comprises an object-oriented operating system for controlling the display device and wherein step D1 comprises the step of:

D1A. creating each of the plurality of view system objects using class information located in the operating system.

16. A method according to claim 10 wherein step D comprises the steps of:

D3. for each of the view system objects, creating a plurality of view objects, each of the plurality of view objects comprising commands for storing screen information into the screen buffer for one window associated with the view system object; and

D4. arranging the plurality of view objects into a view hierarchy.

17. A method according to claim 10 wherein the computer system comprises an object-oriented operating system for controlling the display device and wherein step C comprises the step of:

C1. creating the view system monitor in the object-oriented operating system.

18. A computer program product for controlling a computer system having a display device, a screen buffer and a mechanism for displaying information stored in the screen buffer on the display device to generate a display having a plurality of windows displayed on a desktop background, each of the plurality of windows being assigned to, and displaying screen information generated by, one of a plurality of application programs, the computer program product comprising a computer usable medium having computer readable program code thereon including:

program code for generating a common view system monitor to divide the screen buffer into a plurality of storage areas, each of the plurality of storage areas storing the screen information for one of the plurality of windows;

program code for creating a plurality of view system objects, each of the plurality of view system objects being part of one of the plurality of application programs; and

program code in each of the plurality of view system objects for storing screen information directly into a storage area associated with a window assigned to the application program of which the view system object is a part.

20. A computer program product according to claim 19 wherein the computer system comprises an object-oriented operating system for controlling the display device and wherein the program code for controlling the plurality of application programs comprises program code for creating each of the plurality of view system objects using class information located in the operating system.

24. A computer program product according to claim 18 wherein the program code for creating at least one view object in each of the view system objects comprises:

program code in each of the view system objects for creating a plurality of view objects, each of the plurality of view objects comprising commands for storing screen information into the screen buffer for one window associated with the view system object; and

program code for arranging the plurality of view objects into a view hierarchy.

25. A computer program product according to claim 18 wherein the computer system comprises an object-oriented operating system for controlling the display device and wherein the program code for generating a common view system monitor comprises:

program code for creating the view system monitor object in the object-oriented operating system.

First Hit   Fwd Refs

☐ ▓▓▓▓ Generate Collection ▓▓▓▓

L16: Entry 35 of 39   File: USPT   May 24, 1994

DOCUMENT-IDENTIFIER: US 5315711 A
** See image for Certificate of Correction **
TITLE: Method and apparatus for remotely and centrally controlling a plurality of
host processors

Application Filing Date (1):
19911101

Brief Summary Text (6):
This invention relates to the operation of digital data processing systems. In
particular, this invention shows a system whereby a plurality of digital data
processing systems can be remotely operated from either a single workstation or a
plurality of workstation computers. Still more particularly, the same operator
capabilities, available at the digital data processing system operator's console,
are available at a workstation computer which is coupled to a communications
network which is shared with the operator's console.

Brief Summary Text (11):
One implementation of the easy to use operator interface strategy is the
commercially available SHIELD Friendly Console for 2200 Series Host Processors sold
by the assignee of the present invention. The SHIELD Friendly Console can be used
in place of the traditional Operation Console described earlier. The SHIELD
Friendly Console consists of software running on a personal computer which provides
enhanced display of output messages, a menu driven interface for the generation of
input commands to the Host Processor, and context sensitive help. The combination
of capabilities provided by the SHIELD Friendly Console helps to make the operation
of a complex Host Processor less than an ordeal.

Brief Summary Text (28):
A message requesting that remote operation be enabled is sent from the workstation
to the console over the communications network. Upon receipt of an "Open Console
Window" message from the console, the workstation opens a console window on the
workstation video display, with the console window duplicating the user interface
which is available on the console. After the console window is opened on the
workstation video display, any changes in a console display screen are sent from
the console to the workstation and the changes are reflected in the console window
on the workstation video display.

Brief Summary Text (29):
The following scenarios illustrate the operation of a console window. When a
message is sent from the host processor operating system to the console, the
message is displayed on the console video display in accordance with the console
user interface. At the same time that the console video display is being updated,
the same display changes are sent to a program executing on the workstation. The
workstation program then updates the console window with the same console display
changes.

Brief Summary Text (32):
When using a "menu" interface, one possible scheme is for the program to display a
list of program defined keywords for the user, and the user then has the option of

choosing one of the keywords. Upon selection of a keyword, the program interprets the selection chosen. The selection may be interpreted as a command or may lead to a sub-menu where more keywords are displayed.

Drawing Description Text (20):
FIG. 19 shows the relationship between a CONSOLE SESSION data object, an Open Console Window Function, and a Console Window Opened Demon;

Drawing Description Text (21):
FIG. 20 is a flow chart for the Open Console Window function that is invoked by the Graphical Tool-kit when the Graphical Tool-kit detects that a user has manipulated a Console graphical data object and desires that a Console Window be opened;

Drawing Description Text (24):
FIG. 23 is a flow chart for the Close Console Window function that is invoked by the Graphical Tool-kit when the Graphical Tool-kit detects that a user has manipulated a graphical data object and desires that a Console Window be closed;

Detailed Description Text (34):
The Workstations 70 shown in FIG. 5 are used to remotely control and monitor the Host Processors. In the preferred embodiment, the Workstation consists of a programmable Processing Unit 148 with computing capabilities on a scale comparable to the commercially available Unisys U6000/WS workstation. Because the invention provides for control of multiple Host Processors from a single Workstation a large Video Display Terminal 150 is coupled to the Processing Unit 148 for displaying output from Processing Unit 148. The large Video Display Terminal facilitates interaction with multiple Consoles. While smaller Video Display Terminals could be used to practice the invention, a Video Display Terminal whose display portion measures at least 19 inches is preferred.

Detailed Description Text (70):
In addition to the Host Processor control provided by the Console application programs, in the preferred embodiment the Display Manager provides the capability for a user to individually control each of the Console Windows 170. In particular, where a Console has a Console Window displayed on a plurality of Workstations, the user can select one of the Console Windows from which input will be exclusively received. This provides the capability to stop users from entering conflicting input. The Display Manager communicates with a user through Console Window Control Virtual Screens and Sub-Screens it creates for each of the Terminal Screens. The purpose of the Console Window Control Virtual Screens is to allow the user to view output from the Display Manager processing for Console Window control and enter input to the Display Manager processing for Console Window control. The important point is that the Display Manager separately processes Console Window control input and output for each Console Window.

Detailed Description Text (71):
FIG. 8 shows the relationship between Terminal Screens, Virtual Screens, Sub-Screens, and the Console Window Control Virtual Screens and Sub-Screens. FIG. 8 is the same basic example as shown in FIG. 7. The difference is that a separate set of Console Window Control Virtual Screens, shown as 280, 282, 284, and 286, is maintained for each of the Terminal Screens 250 and 252. Console Window Control Virtual Screens 280 and 282 are mapped to Terminal Screen 250, and Console Window Control Virtual Screens 284 and 286, are mapped to Terminal Screen 252. The Virtual Screens 280, 282, 284, and 286 are merely illustrative of a graphical user interface used by the Display Manager. Those skilled in the art will recognize that alternate user interfaces exist which a Display Manager type program could us to display its control output.

Detailed Description Text (75):
Remote Console Windows 340 and 342 are driven by data supplied from the Terminal

Screens maintained by the Display Manager 84. Block 344 is the Local Console Window of a Terminal Screen which is displayed on the Video Display Terminal 18 that is coupled to the Console Processing Unit 13. Each of the Remote Console Windows 340 and 342 and the Local Console Window 344 can have data generated by a single Console application program, or each may have data generated by a different Console application program.

Detailed Description Text (77):
The Display Manager 84 manages operator input and output for each of the Remote Console Window 340 and 342, and Local Console Window 344. The Display Manager directly processes input and output to the Local Console Window 344 and maintains a Local Terminal Screen 348 data structure whose contents drives the Local Console Window 344. When the Display Manager 84 receives a request to open a Remote Console Window 340, an X Terminal Handler 352 is started to handle input and output to the Remote Console Window 340. The X Terminal Handler 352 creates a Remote Terminal Screen data structure 354 whose contents are displayed in Remote Console Window 340. Likewise, when the Display Manager 84 receives a request to open a Remote Console Window 342, an X Terminal Handler 356 is started to handle input and output to the Remote Console Window 342, and the X Terminal Handler 356 creates a Remote Terminal Screen data structure 358 whose contents are displayed in Remote Console Window 342. X Terminal Handler 360 is shown to illustrate that an arbitrary number of Remote Console Windows could be driven from the Console 12.

Detailed Description Text (88):
If test 562 determines that the Terminal Screen data structure retrieved is associated with a Remote Console Window 340, control path 570 leads to operation 572 sending a request to X Terminal Handler 352 to update the Terminal Screen 354 with information specified in the request. Processing then continues with test 558.

Detailed Description Text (89):
Test 558 checks whether there are any more Terminal Screens to process. If there are more, control path 574 leads to operation 576 retrieving the next Terminal Screen, and control is returned to operation 552 to search for Virtual Screens marked as updated. If test 558 determines that there are no more Terminal Screens to process, control path 580 leads operation 582 clearing the Virtual Screen Updated flags. Processing control is then returned to the main Display Manager processing.
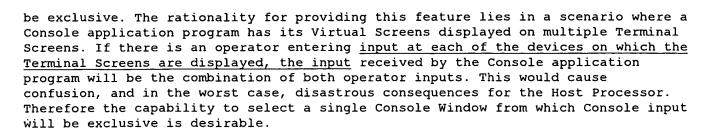
Detailed Description Text (91):
If test 602 is negative, control path 610 leads to test 612 where the Display Manager checks for a close Console Window request. If the test is affirmative, control path 614 leads to operation 616 sending a request to the X Terminal Handler 352 to close the Remote Console Window 340. Processing then returns to the Main Program of FIG. 11.

Detailed Description Text (99):
Once a Remote Console Window 340 has been opened, the X Terminal Handler 352 begins a control loop of processing output requests from the Display Manager 84 and event requests from the X-Server 180 until either the Display Manager or the X-Server closes its communications path with the X Terminal Handler. Operation 710 retrieves a Terminal Screen output request sent by the Display Manager to the X Terminal Handler through the Console Interprocess Communication Handler 86.

Detailed Description Text (113):
The Console Window Control processing as it relates to the present invention coordinates input between the Local Console Window 344 and any Remote Console Windows 340 and 344. In particular, since the present invention involves Console control of multiple Host Processors from multiple Workstations, it is desirable to allow an operator to select a single Console Window from which Console input will

be exclusive. The rationality for providing this feature lies in a scenario where a Console application program has its Virtual Screens displayed on multiple Terminal Screens. If there is an operator entering input at each of the devices on which the Terminal Screens are displayed, the input received by the Console application program will be the combination of both operator inputs. This would cause confusion, and in the worst case, disastrous consequences for the Host Processor. Therefore the capability to select a single Console Window from which Console input will be exclusive is desirable.

Detailed Description Text (117):
Control path 942 leads to test 944 which determines whether the ENTER key has been pressed. When the ENTER key has been pressed, the indication is that a command has been completely entered and can be processed by the appropriate application program. In the case of Console Window Control Menus, when the ENTER key is pressed, the processing application program is the Display Manager. Upon detecting the ENTER key, control path 946 leads to test 948 which determines whether the requested command is to "Take Control". When the Take Control command is entered, control path 950 leads to operation 952 which marks the Terminal Screen as controlling. The effect of this operation is to exclude keyboard input made at devices on which other Terminal Screens are displayed. After marking the Terminal Screen, operation 918 removes the Terminal Screen Control Menu type Virtual Screen from the Terminal Screen's list of Virtual Screens, and operation 920 marks the Terminal Screen as needing to be refreshed. Control is then returned to the Handle Keystroke processing.
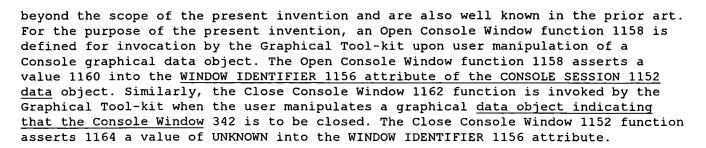
Detailed Description Text (124):
Operation 1056 selects the first Remote Terminal Screen data structure from the list of data structures. Test 1058 checks whether or not a Terminal Screen was found as a result of operation 1056. If a Terminal Screen was found, control path 1060 leads to test 1062 which determines the type of device on which the selected Terminal Screen is displayed. If the display device is a Dumb Terminal, control path 1064 leads to operation 1066 where output of the BEL character to the Dumb Terminal is stopped. If the display device is not a Dumb Terminal, control path 1068 leads to operation 1070 sending an alarm stopped message to the Workstation corresponding to the selected Terminal Screen. The message is sent from the Display Manager 84, through the Console Interprocess Communication Handler 86, to the Network Connection Manager 82 for routing to the appropriate Workstation 70. After completing operations 1066 and 1070, operation 1072 selects the next Terminal Screen from the list of Terminal Screens and control returns to test 1058. When test 1058 finds that there are no more Terminal Screens to process, control path 1074 returns to Handle Keystroke processing.

Detailed Description Text (126):
FIG. 19 shows the relationship between a CONSOLE SESSION data object, an Open Console Window Function, and a Console Window Opened Demon. The specification for the CONSOLE SESSION 1152 data object is defined in the data model. When the processing for the application program 190 is initiated, data objects conforming to the CONSOLE SESSION definition in the data model are created for each Console 12 that is defined for the application 190. The Consoles for which the application 190 creates CONSOLE SESSION data objects are defined by the user prior to initiation of application 190 processing.

Detailed Description Text (128):
Open Console Window 1158 is a function which is invoked by the Graphical Tool-kit 196. The Graphical Tool-kit 196, in combination with the Knowledge Engineering System 192, manages graphical data objects which a user can manipulate through a Keyboard 152 or a Mouse (Not shown in the figures). For each Console 12 that a user has predefined to be part of a configuration, a corresponding graphical data object is created. Upon user action on a Console graphical data object, the Graphical Tool-kit 196 invokes a predetermined function. The details for this processing are

beyond the scope of the present invention and are also well known in the prior art. For the purpose of the present invention, an Open Console Window function 1158 is defined for invocation by the Graphical Tool-kit upon user manipulation of a Console graphical data object. The Open Console Window function 1158 asserts a value 1160 into the <u>WINDOW IDENTIFIER 1156 attribute of the CONSOLE SESSION 1152 data</u> object. Similarly, the Close Console Window 1162 function is invoked by the Graphical Tool-kit when the user manipulates a graphical <u>data object indicating that the Console Window</u> 342 is to be closed. The Close Console Window 1152 function asserts 1164 a value of UNKNOWN into the WINDOW IDENTIFIER 1156 attribute.

<u>Detailed Description Text</u> (130):
FIG. 20 is a flow chart for the Open Console Window function 1158 that is invoked by the Graphical Tool-kit 96 when the Graphical Tool-kit detects that a user has manipulated a Console graphical <u>data object and desires that a Console Window</u> 340 be opened. Operation 1202 retrieves the CONSOLE SESSION data object whose CONSOLE NAME 1152 matches that of the Console graphical data object manipulated. After retrieving the CONSOLE SESSION <u>data object, test 1204 checks whether the WINDOW</u> IDENTIFIER 1156 attribute is UNKNOWN. If the WINDOW IDENTIFIER attribute already has a value, control path 1206 leads to operation 1208 invoking the Graphical Tool-kit to display an error message for the user. After displaying the error message, processing is complete.

<u>Detailed Description Text</u> (133):
FIG. 23 is a flow chart for the Close Console Window function 1162 that is invoked by the Graphical Tool-kit 196 when the Graphical Tool-kit detects that a user has manipulated a graphical <u>data object and desires that a Console Window</u> 340 be closed. Operation 1302 retrieves the CONSOLE SESSION 1152 data object whose CONSOLE NAME 1154 attribute matches that of the name passed to the Close Console Window processing. After the CONSOLE SESSION data object is retrieved, operation 1304 invokes the Graphical Tool-kit 196 processing to close the Window 170 with window identifier that matches the <u>WINDOW IDENTIFIER 1156 attribute in the CONSOLE SESSION 1152 data</u> object. After the Window has been closed, operation 1306 asserts the <u>WINDOW IDENTIFIER 1156 attribute in the CONSOLE SESSION 1152 data</u> object to be UNKNOWN.

☐ ▓▓▓ Generate Collection ▓▓▓

L16: Entry 36 of 39   File: USPT   Apr 19, 1994

DOCUMENT-IDENTIFIER: US 5305435 A
TITLE: Computer windows management system and method for simulating off-screen
document storage and retrieval

Abstract Text (1):
A novel computer display interface which simulates familiar document handling
activities based upon a unique display metaphor representative of a standard office
filing system and work area of a desk. The display system includes multi-windows
which are displayed in a central screen area designated for just active computer
windows. The interface further includes designating the edge areas of the screen
for displaying window tabs of inactive windows. The window tabs are arranged in a
way to simulate an integrated filing system, and to provide a convenient method of
organizing and viewing computer files.

Application Filing Date (1):
19930507

Brief Summary Text (8):
To further aid computer productivity it became desirable to provide a "multi-
tasking" environment in which a user could implement several applications at the
same time. A convenient and effective method of providing multi-tasking has been
provided by the development of multi-window systems. In such systems, different
computer windows, i.e. software applications or data files, may be opened and used
by the user simultaneously. Computer windows appear on the display monitor of a
computer as a virtual window, that is, the contents of a file associated with the
window may be viewed within the area described by the window. Several windows may
appear on the display monitor at the same time and often appear as if they are
layered, one on top of another. For ease of use these multi-window systems often
incorporate the features of icons and pulldown menus found in graphics interfaces.

Brief Summary Text (10):
The screen display is organized in this and other similar interfaces by a series of
windows. A main window contains primary file folders. This window generally
occupies a large portion of the screen. To see the contents of a particular primary
file folder, the user is required to point and click on the icon of that particular
file folder. A pop-up window of icons representing the different files and contents
of the file folder is then displayed. This pop-up window generally overlaps and
hides part of the main window. To actually see the text of a file another point and
click procedure must be done. This results in yet another window appearing on the
screen which completely fills the screen and hides the other vertically stacked
windows. While open, the source of this text file window is not readily apparent,
that is from which file folder it came.

Brief Summary Text (15):
Therefore, in view of the above, it is an object of the present invention to devise
a multi-window computer interface based upon a display metaphor which is intuitive,
well organized and easy to understand.

Brief Summary Text (26):
An advantage of the present invention is that a user interface is provided which

emulates real world desktop activities and filing systems. Therefore, the interface
is easy to learn, since it is intuitive. Indeed, little time would be required to
teach individuals how to use this interface, because the display metaphor relies
upon familiar activities and symbols.

Detailed Description Text (2):
Referring to FIG. 1, a computer screen 10 is shown having multiple windows therein.
The windows represent either application programs or data files. The screen 10 has
a left screen edge 12, a bottom edge 14, a right edge 16 and a top edge 18.
Proximate to and inside L of the screen edges 12-18 is a "screen edge area" of the
screen 10. The screen 10 also has a "central screen area" 20 inside of the edge
area. The central screen area 20 and the edge area together form the display or
visible area of the screen 10. Outside of the screen edges 12-18 is defined a
virtual or nonvisible screen area. The purpose of these visible and nonvisible
screen areas is illustrated and discussed in subsequent figures.

Detailed Description Text (3):
Within the central screen area 20 are several windows. Displaying active windows is
what the central screen area 20 is designated for. For example, a window 22 is for
an application program called ABC. Within this window are five separate windows 61-
65. The windows 61-65 are generally data files, but may be application programs as
well. Similarly application program XYZ, indicated as window 24, has other windows
71-73 therein. The windows 71-73 may also represent data files, but here they
overlap one another. The window 24 also contains a double-star icon 74 which
represents the application XYZ of the window 24. For instance, if application XYZ
was for word processing, the icon could be of something related to word processing
like a keyboard or sheet of typing paper. With a spreadsheet application the icon
could be of a ledger sheet or the like. The icon may also be of a logo of the
software vendor which produced the application.

Detailed Description Text (20):
FIG. 11 is a flow chart of a subroutine for moving a window off the screen, and has
reference to FIGS. 1-3. A first step 180 is to read the active window data stored
in an active window memory 206 (see FIG. 13). The active window data relates to
information defining the various windows that are being displayed and their
positions on the screen. The next step 181 is to select a window. This is done by
doing some predefined action, like placing a cursor within the window. A mouse or
other pointing device may be used for cursor control. At this point a "Fast Tab"
option 182 may be used to "pop" the selected window off the screen. This may be
done by a predefined action, such as double clicking on the window. Otherwise the
window may be moved off the screen by "dragging" it using the cursor. This action
comprises steps 183 and 184.

Detailed Description Text (21):
Once the window is off the screen, then a file tab is created for that window. If
there is a particular icon associated with that window then the icon is used on the
window tab, step 185. If there is no icon, then a name for the window is used, step
186. If there is no name, then one is assigned, step 187. The name may be
automatically created or input by the user. Typically, though, file names are asked
for or created by the application program which was used to create the file
originally. Once the identifying information for the tab is determined, e.g. icon
or name, that information is added to a Window Tabs Management Table 210 (see FIG.
13) in step 188. Finally the tab is displayed in step 189.

Detailed Description Text (22):
FIG. 12 is a flow chart of a subroutine for bringing a window which is off screen
back onto the screen. This operation relates to FIGS. 4-6. The operation begins
with the step 191 of reading the data stored in the Window Tabs Management Table
210 (see FIG. 13). This is followed by the step 192 of selecting a particular tab,
by placing a cursor on it. At this point a "Fast Restore" option, step 193, may be

selected by some predefined action, such as double clicking. If the "Fast Restore" option is not used, then the tab is "dragged" onto the central screen area, steps 194 and 195. Once the tab is in the central screen area, the corresponding window file is moved to the active memory 206 (see FIG. 13), step 196. At this point the window may be displayed as it is pulled onto the screen, step 197. A further option is to select a tab for a window within the main window, step 198, and then to repeat steps 192 to 197.

Detailed Description Text (23):
FIG. 13 illustrates a computer system for operating a computer display screen in the manner described above. The system comprises an input device, such as a keyboard 200 and a pointing device, such as a mouse 202. These devices are connected to a CPU 204 via a data bus 218, which represents the various cables used to connect the different system components together.

Detailed Description Text (24):
The memory of the system includes an Active Window Memory 206 for storing information pertaining to active windows, which are those windows being displayed. An Inactive Window Memory 208 is for storing information related to inactive windows. A mass storage device, such as a hard disk, could be used for this purpose. In addition, a Window Tabs Management Table 210 is maintained. This table stores information for displaying the window tabs around the edges of the screen. This information would include position, order, pointers to the corresponding window in the Inactive Window Memory 208, and pointers to associated tabs. Indeed, the Window Tabs Management Table 210 acts as a directory for all of the computer files and windows.

Detailed Description Text (25):
Display data is transferred to a bit map processor 212 in accordance with the Active Window Memory 206 and the Window Tabs Management Table 210. The bit map processor 212 expands the display data as a bit map in a frame memory 214. A controller, not shown, then sequentially reads out the content of the frame memory 214 and outputs this data to a CRT 216.

CLAIMS:

1. A computer windows management system comprising:

a display screen having a central display area and having an edge area between the central display area and a boundary of the display screen, the central display area being designated for displaying computer windows;
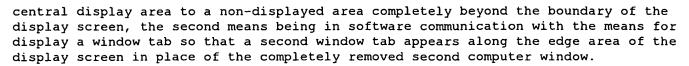
means for displaying a window tab upon removal of a computer windows from the display screen that the window tab has an appearance of being attached to the computer window and such that the window tab is arranged along the edge area of the display screen;

cursor means under operator control for selecting among a plurality of windows tabs to display desired computer windows on the central display area of the display screen and for selecting computer windows to be removed from the display screen;

first means responsive to selection of a first window tab by the cursor means for manipulating a first computer window in a manner to simulate popping of the first computer window to the central display area from a non-displayed area completely beyond the boundary of the display screen in an absence of a user dragging the first computer window using the cursor means; and

second means responsive to the cursor means for manipulating a second computer window in a manner to gradually remove the second computer window completely from the display screen by simulating sliding of the second computer window from the

central display area to a non-displayed area completely beyond the boundary of the display screen, the second means being in software communication with the means for display a window tab so that a second window tab appears along the edge area of the display screen in place of the completely removed second computer window.

5. A method of managing computer windows for displaying stored data comprising:

providing a display screen having a plurality of window tabs for identifying computer windows, including a first window tab for identifying a first computer window, each window tab extending from a boundary of the display screen,

initiating an open command to open the first computer window for viewing stored data on the display screen, wherein initiating the open command includes directing a cursor to a window tab for identifying the first computer window, and

in response to the open command, simulating dragging a rectangular document from a non-displayed area completely beyond the boundary of the display screen without a user dragging the first computer window using a cursor device by displaying the first computer window on the display screen such that a first edge of the first computer window enters at a boundary of the display screen and moves in a direction away from the boundary of the display screen, while parallel second and third edges of the first computer window remain a fixed distance apart.

7. The method of claim 6 further comprising displaying stored data within said first computer window during the movement of the first edge, the stored data moving along the display screen simultaneously with said first edge.

9. A method of managing computer windows of a computer system having a display screen, each computer window having a capability of being activated wherein information contained therein is displayed on the display screen, each computer window further having a capability of being inactivated wherein the computer window and information therein is removed from the display screen, the method comprising:

operating the computer system such that the display screen is organized into a central display area and an edge display area, including reserving the central display area from activating computer windows and further including displaying window tabs within the edge display area for identifying particular computer windows which have been inactivated, each window tab having an appearance of being coupled to each one of the computer windows;

generating computer files within a memory of the computer system such that each computer file is viewable within one of the computer windows;

viewing a first computer file, including generating an open command by designating a first window tab within the edge area;

activating a first computer window in response to the open command, including displaying the first computer file, further including simulating moving the first window tab from the edge display area toward the central display area from a non-displayed location entirely beyond an edge of the display screen into the central display area without a user dragging the first computer window using a cursor device; and

inactivating the first computer window including simulating moving of the first computer window such that the first computer window appears to move entirely beyond the edge of the display screen so that the window and information therein is removed from the display, and such that first window tab is displayed along the edge display area in place thereof.

First Hit   Fwd Refs
Search Forms

Search Results   ☐ ▓▓▓ Generate Collection ▓▓▓

Help

Us6Searches38 of 39   File: USPT   Jul 30, 1991

Preferences

Logout
DOCUMENT-IDENTIFIER: US 5036315 A
TITLE: Simultaneous display of interleaved windowed video information from multiple
asynchronous computers on a single video monitor

Abstract Text (1):
A first computer workstation running windowed display system and applications
programs produces a first video signal of a composite, windowed, display plus a
first data signal indicating the dedications, positions, and sizes of all windows.
A second computer asynchronously digitally communicates a second display signal to
a communications controller and then to a display controller. The display
controller also receives the first data signal plus the sync pulse of the first
video signal. It produces, from the second display signal, a second video display
signal that (i) is synchronized with the first video signal and (ii) contains
display information positioned and scaled to be within a window dedicated to the
second computer. It produces a switch control signal indicating the portions of
each raster scan of the synchronized video signals that are inside and outside the
dedicated window. The switch control signal gates in a video switch either the
first or the second video signal to a video monitor. The video monitor displays
windows showing information from the first computer plus a dedicated window showing
information from the second computer.

Application Filing Date (1):
19880906

Brief Summary Text (7):
A second standard computer component preferably used in the system of the present
invention is a computer of the IBM Corporation, such as a mainframe computer of the
type System 360.TM.or System 370.TM.(trademarks of IBM Corporation). The IBM.RTM.
(registered trademark of IBM Corporation) mainframe computer typically runs CAD/CAM
software applications programs such as CADAM.RTM. (registered trademark of CADAM,
Inc.) or CATIA.RTM. (registered trademark of Dassault Systemes). These mainframe
computers and accompanying software programs support graphic displays at graphics
terminals, such as at terminal type 5080 available from IBM Corporation, in the
performance of CAD/CAM tasks. It will not be necessary in the system of the present
invention that the second computer--normally the IBM System 360 or System 370
mainframe--should be running CAD/CAM application software. More mundane software
programs such as those performing word processing and other business-related tasks
are fully suitable. Any program providing a display will be seen to communicate
with a video monitor that is also communicating with the first computer by action
of the system of the present invention. The situation wherein an existing
VAXstation and an existing IBM mainframe computer are both typically running
CAD/CAM applications software merely represents one normal and preferred
application of the system of the present invention.

Brief Summary Text (9):
For the purposes of the present invention, it should be understood that the
existing DSCC (and the DS 1082GXP Display Controller to which it is connected, next
discussed) buffer information from a mainframe computer. This information later
results in a video display. This buffering effectively permits a decoupling of the
signals driving a video monitor, and also of the resulting displays on the video

monitor, from any required synchronization with the communication channels of the mainframe computers to which the DSCC and Display Controller connect.

Brief Summary Text (10):
A fourth device used, in a modified form, in the system of the present invention is the Display Controller component of the DesignSet.TM. 1082GXP.TM. (trademarks of Spectragraphics Corporation) high-performance graphic system product of Spectragraphics Corporation. The DS 1082GXP graphics system includes, in addition to a Display Controller component, a 16 inch video monitor, a keyboard, and a mouse. These additional, peripheral, components may also optionally be employed within the system of the present invention. If they are so employed then they are in lieu of or supplementary to the equivalent components which normally accompany the VAXstation.

Brief Summary Text (11):
The DS 1082GXP graphics system serves to emulate the 5080 terminal, and optionally additionally the 3270 terminal, of IBM Corporation. For the purposes of the present invention, it need only be understood that the Display Controller of the DS 1082GXP has a digital interface to the DSCC. In response to commands and information received across this interface it produces a standard Red Green Blue (RGB) video output signal which drives the presentation of displays upon a standard video monitor.

Brief Summary Text (12):
Both the DSCC and the Display Controller component of the DS 1082GXP are controlled in operation by resident firmware, or microcode, programs. No modification is required to the firmware control program of the DSCC to support its use in the system of the present invention. A modification and addition to the firmware control program of the Display Controller, as well as a modification to the hardware itself, will be seen to be required to support use of the Display Controller within the system of the present invention.

Brief Summary Text (21):
The present invention contemplates a digital computer graphics and communications system wherein two or more computers each cause the simultaneous display of information each within one or more windows appearing upon a single video monitor. Normal manually-directed cursor movement between windows accords communication, typically via keyboard and/or mouse, with that computer that is controlling a particular window within which the cursor is placed. The input and output peripherals of one workstation thus suffice to communicate with two or more computers.

Brief Summary Text (24):
The digital computer graphics and communications system in accordance with the present invention intentionally combines and couples selective hardware and software components of existing systems in order to realize, in combination and as occasionally modified, a new function. A first antecedent system to the system of the present invention is a computer workstation that includes a first, system-window-managing, computer. This first computer runs a software program that causes production of a video display signal that provides a windowed display upon a video monitor. Input peripherals such as a keyboard and a mouse communicate with the first computer.

Brief Summary Text (25):
A second antecedent system to the system of the present invention is a computer system that includes a second, host, computer. This second computer runs software producing a display signal that would generate a (typically non-windowed) display if the computer were to be directly connected to a graphics terminal. The display signal from the second computer is digital and not video. It is routed across parallel interface cables to a communications controller, and not directly to a

First Hit   Fwd Refs

☐ ▨▨▨ Generate Collection ▨▨▨

L24: Entry 1 of 31   File: USPT   Jul 10, 2001

DOCUMENT-IDENTIFIER: US 6257981 B1
** See image for Certificate of Correction **
TITLE: Computer network for controlling and monitoring gaming devices

Application Filing Date (1):
19970902

Brief Summary Text (3):
Networked gaming devices are know in the art. Interconnecting a plurality of gaming
devices such as slot machines via a computer network to a central computer provides
many advantages. The primary advantage of networked gaming devices is the ability
to extract accounting data from the individual gaming devices as well as providing
player tracking. An as example of a data collection system is described in U.S.
Pat. No. 4,283,709 issued to Lucero et al. Network systems such as described in
Lucero et al. allow the central host computer to monitor the usage and payout,
collectively known as audit data, of the individual gaming devices. This audit data
includes data related to the number of coins or tokens inserted into the device,
the number of times the device has been played, the amount paid in raises, the
number and the type of jackpots paid by the machine, the number of door openings,
etc. The host computer can then compile an accounting report based on the audit
data from each of the individual gaming devices. This report can then be used by
management, for example, to assess the profitability of the individual gaming
devices.

Brief Summary Text (14):
The system includes a plurality of gaming devices or machines connected to an
associated floor controller over a network. The system includes one or more of said
floor controllers. The floor controllers are interconnected by a high-speed
network, such as an Ethernet network, to i s a database where accounting and player
tracking data is stored. The system can also include pit terminals and/or fill and
jackpot processing terminals. Each promotion involves sending a reconfiguration
command from the floor controller to a gaming device that has been selected to be
part of a given promotion over the associated network. Upon receipt of the
reconfiguration command, the gaming device reconfigures its payout schedule in
accordance with the received reconfiguration command. In the-preferred embodiment,
this reconfiguration includes activating a bonus payout schedule. A partial list of
the promotions according to the invention include, but are not limited to: a
multiple jackpot wherein the gaming device reconfigures its payout to be a multiple
of its default payout schedule; a bonus jackpot wherein the gaming device
reconfigures its payout schedule to payout an additional bonus amount when certain
conditions are met; and a progressive jackpot wherein two or more gaming devices
are combined in a progressive jackpot having a progressive jackpot payout schedule.
In addition to these, many other promotions are possible by the above-described
system for controlling and monitoring a plurality of gaming devices.

Detailed Description Text (46):
A system for operating a plurality of gaming devices is shown generally at 10 in
FIG. 1. The system, hereinafter described, monitors and reconfigures a plurality of
gaming devices or machines 12-16 and 22-26. The system includes the following
capabilities: remote reconfiguration, accounting data extraction, integrate&player

tracking, and cashless play. Remote reconfiguration includes sending a reconfiguration command from a host computer to one or more of the gaming devices. The gaming devices, on receiving a reconfiguration command, will reconfigure its jackpot payout schedule in accordance with the reconfiguration command.

Detailed Description Text (108):
The display controller 174 interfaces with the driver circuit 186 by a plurality of signal lines 190. These signal lines transmit the standard driver interface signals to the driver circuit 186. These signals include: a clock signal CLOCK, serial input data signal SDATA, a frame signal FRAME, a strobe signal STROBE, two output enable signals OE1/ and OE2/, a column clock signal COL CLOCK, and a column output enable signal COL OE/. These signals have well known functions in the display art and are therefor not discussed in detail. The signal names having a "/" represent active low signals while all other signals are active high. The display controller 174 generates these signals in the required sequence in order to serially clock the reformatted display data to the driver circuit. One of ordinary skill in the art could program the display controller 176 to generate these signals in order to display the desired message on the VFD 184 based on the foregoing description.

Detailed Description Text (111):
The display 102 further includes a discrete input section 198. The discrete input section 198 is an interface between the discrete outputs of a gaming device and the display controller 174 much in the same way that the discrete machine interface 72 allows the data communication node to interface with a gaming device. Although in the preferred embodiment the discrete input section is unconnected to any discrete machine inputs, the discrete input section 198 allows the display 102 to operate as a stand-alone module for gaming devices in certain configurations. The discrete input section provides discrete input signals from an external device to the display controller 174 over a bus 200. The discrete input section 198 includes opto-isolator circuits such as part number TLP620 manufactured by Toshiba Corporation of Tokyo, Japan which provide single-ended input signals to the display controller 174.

Detailed Description Text (129):
As shown in FIG. 1, the floor controller is directly connected to both the high speed network 38 and a plurality of gaming devices. The floor controller is responsible for monitoring the activity of each of the gaming devices connected thereto and reporting this activity to the database 32. In addition, the floor controller is responsible for transmitting a reconfiguration command to a selected one or more of the gaming devices during certain bonus conditions. These conditions will be described in detail in the operation section below.

Detailed Description Text (137):
A partial list of the promotions according to the invention include, but are not limited to: a multiple jackpot wherein the gaming device reconfigures its payout to be a multiple of its default payout schedule; a bonusjackpot wherein the gaming device reconfigures its payout schedule to payout an additional bonus amount when certain conditions are met; and a progressive jackpot wherein two or more gaming devices are combined in a progressive jackpot having a progressive jackpot payout schedule. In addition to these, many other promotions are possible by the above-described system for controlling and monitoring a plurality of gaming devices.

First Hit   Fwd Refs

☐ [Generate Collection]

L24: Entry 2 of 31   File: USPT   Jun 12, 2001

DOCUMENT-IDENTIFIER: US 6244958 B1
** See image for Certificate of Correction **
TITLE: Method for providing incentive to play gaming devices connected by a network to a host computer

Application Filing Date (1):
19960625

Brief Summary Text (5):
Networked gaming devices are known in the art. Interconnecting a plurality of gaming devices, such as slot machines, via a computer network to a central computer provides many advantages. Such advantages include compiling and auditing data related to the amount of coins received by the gaming devices as well as the amount paid to players of the devices.

Detailed Description Text (55):
A system for operating a plurality of gaming devices is shown generally at 10 in FIG. 1. The system, hereinafter described, monitors and reconfigures a plurality of gaming devices or machines 12-16 and 22-26. The system includes the following capabilities: remote reconfiguration, accounting data extraction, integrated player tracking, and cashless play. Remote reconfiguration includes sending a reconfiguration command from a host computer to one or more of the gaming devices. The gaming devices, on receiving a reconfiguration command, will reconfigure its jackpot payout schedule in accordance with the reconfiguration command.

Detailed Description Text (118):
The display controller 174 interfaces with the driver circuit 186 by a plurality of signal lines 190. These signal lines transmit the standard driver interface signals to the driver circuit 186. These signals include: a clock signal CLOCK, serial input data signal SDATA, a frame signal FRAME, a strobe signal STROBE, two output enable signals OE1/and OE2/, a column clock signal COL CLOCK, and a column output enable signal COL OE/. These signals have well known functions in the display art and are therefor not discussed in detail. The signal names having a "/" represent active low signals while all other signals are active high. The display controller 174 generates these signals in the required sequence in order to serially clock the reformatted display data to the driver circuit. One of ordinary skill in the art could program the display controller 176 to generate these signals in order to display the desired message on the VFD 184 based on the foregoing description.

Detailed Description Text (121):
The display 102 further includes a discrete input section 198. The discrete input section 198 is an interface between the discrete outputs of a gaming device and the display controller 174 much in the same way that the discrete machine interface 72 allows the data communication node to interface with a gaming device. Although in the preferred embodiment the discrete input section is unconnected to any discrete machine inputs, the discrete input section 198 allows the display 102 to operate as a stand-alone module for gaming devices in certain configurations. The discrete input section provides discrete input signals from an external device to the display controller 174 over a bus 200. The discrete input section 198 includes opto-isolator circuits such as part number TLP620 manufactured by Toshiba

Corporation of Tokyo, Japan which provide single-ended input signals to the display
controller 174.

Detailed Description Text (139):
As shown in FIG. 1, the floor controller is directly connected to both the high
speed network 38 and a plurality of gaming devices. The floor controller is
responsible for monitoring the activity of each of the gaming devices connected
thereto and reporting this activity to the database 32. In addition, the floor
controller is responsible for transmitting a reconfiguration command to a selected
one or more of the gaming devices during certain bonus conditions. These conditions
will be described in detail in the operation section below.

Detailed Description Text (148):
A partial list of the promotions according to the invention include, but are not
limited to: a multiple jackpot wherein the gaming device reconfigures its payout to
be a multiple of its default payout schedule; a bonus jackpot wherein the gaming
device reconfigures its payout schedule to payout an additional bonus amount when
certain conditions are met; and a progressive jackpot wherein two or more gaming
devices are combined in a progressive jackpot having a progressive jackpot payout
schedule. In addition to these, many other promotions are possible by the above-
described system for controlling and monitoring a plurality of gaming devices.

□ [Generate Collection]

L24: Entry 8 of 31   File: USPT   Mar 2, 1999

DOCUMENT-IDENTIFIER: US 5876284 A
TITLE: Method and apparatus for implementing a jackpot bonus on a network of gaming devices

Application Filing Date (1):
19960513

Brief Summary Text (3):
Networked gaming devices are known in the art. Interconnecting a plurality of gaming devices such as slot machines via a computer network to a central computer provides many advantages. Some advantages of a network for operating networked gaming devices, like that disclosed in U.S. application Ser. No. 08/322,172, filed Oct. 12, 1994, assigned to the assignee of the present application, include the ability to extract accounting data from the individual gaming devices, to track players and to operate bonus promotions and progressive jackpots.
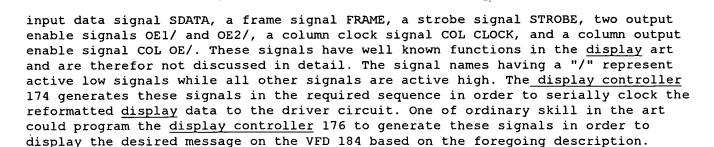
Brief Summary Text (16):
The system includes a plurality of gaming devices or machines connected to an associated floor controller over a network. The system includes one or more of said floor controllers. The floor controllers are interconnected by a high-speed network, such as an Ethernet network, to a database where accounting and player tracking data is stored. The system can also include pit terminals and/or fill and jackpot processing terminals. Each promotion involves sending a reconfiguration command from the floor controller to a gaming device that has been selected to be part of a given promotion over the associated network. Upon receipt of the reconfiguration command, the gaming device reconfigures its payout schedule in accordance with the received reconfiguration command. In the preferred embodiment, this reconfiguration includes activating a bonus payout schedule. A partial list of the promotions according to the invention include, but are not limited to: a multiple jackpot wherein the gaming device reconfigures its payout to be a multiple of its default payout schedule; and a bonus jackpot wherein the gaming device reconfigures its payout schedule to payout an additional bonus amount when certain conditions are met. In addition to these, many other promotions are possible by the above-described system for controlling and monitoring a plurality of gaming devices.

Detailed Description Text (48):
A system for operating a plurality of gaming devices is shown generally at 10 in FIG. 1. The system, hereinafter described, monitors and reconfigures a plurality of gaming devices or machines 12-16 and 22-26. The system includes the following capabilities: remote reconfiguration, accounting data extraction, integrated player tracking, and cashless play. Remote reconfiguration includes sending a reconfiguration command from a host computer to one or more of the gaming devices. The gaming devices, on receiving a reconfiguration command, will reconfigure its jackpot payout schedule in accordance with the reconfiguration command.

Detailed Description Text (110):
The display controller 174 interfaces with the driver circuit 186 by a plurality of signal lines 190. These signal lines transmit the standard driver interface signals to the driver circuit 186. These signals include: a clock signal CLOCK, serial

input data signal SDATA, a frame signal FRAME, a strobe signal STROBE, two output enable signals OE1/ and OE2/, a column clock signal COL CLOCK, and a column output enable signal COL OE/. These signals have well known functions in the <u>display</u> art and are therefor not discussed in detail. The signal names having a "/" represent active low signals while all other signals are active high. The <u>display controller</u> 174 generates these signals in the required sequence in order to serially clock the reformatted <u>display</u> data to the driver circuit. One of ordinary skill in the art could program the <u>display controller</u> 176 to generate these signals in order to <u>display</u> the desired message on the VFD 184 based on the foregoing description.

Detailed Description Text (113):
The display 102 further includes a discrete input section 198. The discrete input section 198 is an interface between the discrete outputs of a gaming device and the display controller 174 much in the same way that the discrete machine interface 72 allows the data <u>communication node to interface</u> with a gaming device. Although in the preferred embodiment the discrete input section is unconnected to any discrete machine inputs, the discrete input section 198 allows the display 102 to operate as a stand-alone module for gaming devices in certain configurations. The discrete input section provides discrete input signals from an external device to the display controller 174 over a bus 200. The discrete input section 198 includes opto-isolator circuits such as part number TLP620 manufactured by Toshiba Corporation of Tokyo, Japan which provide single-ended input signals to the display controller 174.

Detailed Description Text (131):
As shown in FIG. 1, the floor controller is directly connected to both the high speed network 38 and a <u>plurality of gaming devices</u>. The floor controller is responsible for monitoring the activity of each of the gaming devices connected thereto and reporting this activity to the database 32. In addition, the floor controller is responsible for transmitting a reconfiguration command to a selected one or more of the gaming devices during certain bonus conditions. These conditions will be described in detail in the operation section below.

Detailed Description Text (147):
A partial list of the promotions according to the invention include, but are not limited to: a multiple jackpot wherein the gaming device reconfigures its payout to be a multiple of its default payout schedule; a bonus jackpot wherein the gaming device reconfigures its payout schedule to payout an additional bonus amount when certain conditions are met; and a progressive jackpot wherein two or more gaming devices are combined in a progressive jackpot having a progressive jackpot payout schedule. In addition to these, many other promotions are possible by the above-described system for controlling and monitoring a <u>plurality of gaming devices</u>.

CLAIMS:

15. A system for operating a <u>plurality of gaming devices</u> interconnected by a computer network to a host computer comprising:

an output port associated with said host computer and connected to said network, said port being operable to send a signal generated by said computer on said network which configures at least one gaming device for operation in a bonus mode,

an interface unit contained in said gaming device and operably connected to said network, said interface device responding to said signal by configuring said gaming device to pay jackpots according to a bonus payout schedule; and

an indicator connected to said network, said indicator comprising a speaker and a light for simulating a storm and generating indicia intelligible to a player of said gaming device when said bonus mode is effective.

First Hit   Fwd Refs

☐ ░░ Generate Collection ░░

L24: Entry 5 of 31   File: USPT   Jul 18, 2000

DOCUMENT-IDENTIFIER: US 6092202 A
TITLE: Method and system for secure transactions in a computer system

Application Filing Date (1):
19980522

Detailed Description Text (43):
The above described model of interface communication protocol for the enforcement
of the interface 134 functionality highlights the distributed processing advantage
provided by the secure computing environment 104. As demonstrated, the computer 114
has no direct access to the sensitive data throughout the entire electronic
transaction process. ·

Detailed Description Text (45):
As illustrated in FIG. 6, the security co-processor 400 includes a processor 410
(e.g. a 32 bit reduced instruction set controller (RISC)). The processor 410 is
coupled to an ISO7816 smart card interface 414 for interfacing to smart cards 436
which conform to the ISO7816 standard. The processor 410 is further coupled to a
keyboard/keypad interface 416 and a display controller interface 418 for
interfacing with the trusted input device 438 and the trusted display 440,
respectively. A microprocessor support 420 coupled to the processor 410 includes
components, such as interrupt controller and timers, needed for the operation of
the processor 420.

CLAIMS:

10. The system of claim 9, in which the security co-processor includes:

a microprocessor;

a microprocessor support coupled to the microprocessor;

a display interface coupled to the microprocessor, the display interface for
providing the visual feedback and the true transaction information to the trusted
display;

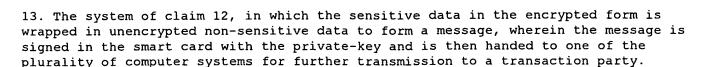first means interfacing between the microprocessor and the trusted input means;

second means interfacing between the microprocessor and the smart card interface;

a memory coupled to the microprocessor;

an external memory interface coupled to the microprocessor; the external memory
interface for interfacing with an external memory;

a cryptographic controller unit coupled to the microprocessor, the cryptographic
controller unit for providing data in encrypted form; and

third means for interfacing between the microprocessor and a plurality of computer
systems.

13. The system of claim 12, in which the sensitive data in the encrypted form is wrapped in unencrypted non-sensitive data to form a message, wherein the message is signed in the smart card with the private-key and is then handed to one of the plurality of computer systems for further transmission to a transaction party.

17. A system for secure transactions in a host computer, comprising:

a security co-processor, the security co-processor including a processor, a processor support coupled to the processor; a display interface coupled to the processor, first interface means for receiving trusted input, the first interface means being coupled to the processor, smart card interface means coupled to the processor, a memory coupled to the processor, an external memory interface coupled to the processor, a cryptographic unit coupled to the processor, and second interface means coupled to the processor, the second interfacing means for interfacing with a plurality of computer systems;

an interface for interfacing between the security co-processor and the host computer, the interface including an interface communication protocol for restricting access by the host computer to data passing through the security co-processor, wherein the interface is coupled to the second interface means;

means for providing trusted input coupled to the security co-processor via the first interface means;

a smart card interface coupled to the smart card interface means of the security co-processor, the smart card interface for interfacing between the security co-processor and smart cards; and

a trusted display coupled to the display interface of the security co-processor for providing a visual feedback and true transaction information, wherein secure transaction processing is performed locally in

the security co-processor and non-secure transaction processing is performed in the host computer system.

23: The method of claim 22, wherein the security co-processor includes:

a processor;

a processor support coupled to the processor;

a display interface coupled to the processor;

first interfacing means for interfacing with one of a keyboard and a keypad, the first interfacing means being coupled to the processor;

a smart card interface coupled to the processor;

a memory coupled to the processor;

an external memory interface coupled to the processor;

a cryptographic unit coupled to the processor; and

second interfacing means for interfacing with a plurality of computer systems, the second interfacing means being coupled to the processor.

30. The system of claim 28, wherein the executing step f) further includes the steps of:

f4) transferring the message to a smart card via the smart card interface for signing;

f5) signing the message in the smart card with a private-key; and

f6) handing the message to one of the plurality of computer systems for further transmission to a transaction party.

First Hit   Fwd Refs

☐ ▓ Generate Collection ▓

L24: Entry 10 of 31   File: USPT   Nov 17, 1998

DOCUMENT-IDENTIFIER: US 5836817 A
** See image for Certificate of Correction **
TITLE: Method and apparatus for operating networked gaming devices

Application Filing Date (1):
19950606

Brief Summary Text (3):
Networked gaming devices are know in the art. Interconnecting a plurality of gaming
devices such as slot machines via a computer network to a central computer provides
many advantages. The primary advantage of networked gaming devices is the ability
to extract accounting data from the individual gaming devices as well as providing
player tracking. An example of a data collection system is described in U.S. Pat.
No. 4,283,709 issued to Lucero et al. Network systems such as described in Lucero
et al. allow the central host computer to monitor the usage and payout,
collectively known as audit data, of the individual gaming devices. This audit data
includes data related to the number of coins or tokens inserted into the device,
the number of times the device has been played, the amount paid in raises, the
number and the type of jackpots paid by the machine, the number of door openings,
etc. The host computer can then compile an accounting report based on the audit
data from each of the individual gaming devices. This report can then be used by
management, for example, to assess the profitability of the individual gaming
devices.

Brief Summary Text (14):
The system includes a plurality of gaming devices or machines connected to an
associated floor controller over a network. The system includes one or more of said
floor controllers. The floor controllers are interconnected by a high-speed
network, such as an Ethernet network, to a database where accounting and player
tracking data is stored. The system can also include pit terminals and/or fill and
jackpot processing terminals. Each promotion involves sending a reconfiguration
command from the floor controller to a gaming device that has been selected to be
part of a given promotion over the associated network. Upon receipt of the
reconfiguration command, the gaming device reconfigures its payout schedule in
accordance with the received reconfiguration command. In the preferred embodiment,
this reconfiguration includes activating a bonus payout schedule. A partial list of
the promotions according to the invention include, but are not limited to: a
multiple jackpot wherein the gaming device reconfigures its payout to be a multiple
of its default payout schedule; a bonus jackpot wherein the gaming device
reconfigures its payout schedule to payout an additional bonus amount when certain
conditions are met; and a progressive jackpot wherein two or more gaming devices
are combined in a progressive jackpot having a progressive jackpot payout schedule.
In addition to these, many other promotions are possible by the above-described
system for controlling and monitoring a plurality of gaming devices.

Detailed Description Text (46): ·
A system for operating a plurality of gaming devices is shown generally at 10 in
FIG. 1. The system, hereinafter described, monitors and reconfigures a plurality of
gaming devices or machines 12-16 and 22-26. The system includes the following
capabilities: remote reconfiguration, accounting data extraction, integrated player

tracking, and cashless play. Remote reconfiguration includes sending a reconfiguration command from a host computer to one or more of the gaming devices. The gaming devices, on receiving a reconfiguration command, will reconfigure its jackpot payout schedule in accordance with the reconfiguration command.

Detailed Description Text (108):
The display controller 174 interfaces with the driver circuit 186 by a plurality of signal lines 190. These signal lines transmit the standard driver interface signals to the driver circuit 186. These signals include: a clock signal CLOCK, serial input data signal SDATA, a frame signal FRAME, a strobe signal STROBE, two output enable signals OE1/and OE2/, a column clock signal COL CLOCK, and a column output enable signal COL OE/. These signals have well known functions in the display art and are therefor not discussed in detail. The signal names having a "/" represent active low signals while all other signals are active high.

Detailed Description Text (112):
The display 102 further includes a discrete input section 198. The discrete input section 198 is an interface between the discrete outputs of a gaming device and the display controller 174 much in the same way that the discrete machine interface 72 allows the data communication node to interface with a gaming device. Although in the preferred embodiment the discrete input section is unconnected to any discrete machine inputs, the discrete input section 198 allows the display 102 to operate as a stand-alone module for gaming devices in certain configurations. The discrete input section provides discrete input signals from an external device to the display controller 174 over a bus 200. The discrete input section 198 includes opto-isolator circuits such as part number TLP620 manufactured by Toshiba Corporation of Tokyo, Japan which provide single-ended input signals to the display controller 174.

Detailed Description Text (130):
As shown in FIG. 1, the floor controller is directly connected to both the high speed network 38 and a plurality of gaming devices. The floor controller is responsible for monitoring the activity of each of the gaming devices connected thereto and reporting this activity to the database 32. In addition, the floor controller is responsible for transmitting a reconfiguration command to a selected one or more of the gaming devices during certain bonus conditions. These conditions will be described in detail in the operation section below.

Detailed Description Text (138):
A partial list of the promotions according to the invention include, but are not limited to: a multiple jackpot wherein the gaming device reconfigures its payout to be a multiple of its default payout schedule; a bonus jackpot wherein the gaming device reconfigures its payout schedule to payout an additional bonus amount when certain conditions are met; and a progressive jackpot wherein two or more gaming devices are combined in a progressive jackpot having a progressive jackpot payout schedule. In addition to these, many other promotions are possible by the above-described system for controlling and monitoring a plurality of gaming devices.

CLAIMS:

1. A method of operating gaming devices interconnected by a computer network to a host computer comprising:

selecting a plurality of the gaming devices;

using the network to track the amount of money played on the selected gaming devices;

allocating a predetermined percentage of the money played to a bonus pool;

issuing a command over the network including data establishing criteria to cause a bonus to be paid from the pool via one of said selected gaming devices upon the occurrence of a predetermined event;

storing the command in a memory connected to a controller associated with only one of the gaming devices;

transmitting data indicative of gaming device activity from the gaming device to the controller;

transmitting a pay command from the controller to the gaming device upon the occurrence of the predetermined event; and

paying the bonus via the gaming device responsive to receipt of the pay command.

9. The method of claim 1 wherein the step of using the network to track the amount of money played on the selected gaming devices occurs after the step of selecting a plurality of the gaming devices.

13. The method of claim 1 wherein said host computer includes a user-operated input device and wherein selecting a plurality of the gaming devices comprises effecting action the input device.

21. A method of operating gaming devices interconnected by a computer network to a host computer comprising:

selecting a plurality of the gaming devices;

issuing a command over the network including data establishing criteria to cause a bonus to be paid via one of said selected gaming devices upon the occurrence of a predetermined event;

storing the command in a memory connected to a controller associated with only one of the gaming devices;

transmitting data indicative of gaming device activity from the gaming device to the controller;

transmitting a pay command from the controller to the gaming device upon the occurrence of the predetermined event; and

paying the bonus via the gaming device responsive to receipt of the pay command.

24. A method of operating gaming devices interconnected by a computer network to a host computer comprising:

selecting a plurality of the gaming devices;

using the network to track the amount of money played on the selected gaming devices;

allocating a predetermined percentage of the money played to a bonus pool;

initiating a bonus period after the bonus pool exceeds a predetermined level;

providing data establishing criteria to cause a bonus to be paid from the pool via one of said selected gaming devices upon the occurrence of a predetermined event;

storing the data in a memory connected to a controller associated with only one of the gaming devices;

transmitting data indicative of gaming device activity from the gaming device to the controller;

initiating the bonus period;

transmitting a pay command from the controller to the gaming device upon the occurrence of the predetermined event; and

paying the bonus via the gaming device responsive to receipt of the pay command.

☐ | Generate Collection |

L24: Entry 7 of 31   File: USPT   May 4, 1999

DOCUMENT-IDENTIFIER: US 5901067 A
TITLE: System for interactively selecting and activating groups of electrically
powered devices

Abstract Text (1):
A system for interactively selecting and activating groups of electrically powered
devices such as computer and computer peripheral devices. A preferred system
includes computer and computer peripheral devices, an interactive video display and
a controller. The computer and computer peripheral devices include a printer and at
least two computers, each of the computers including a printer port. The computers
include a controller interface computer which executes a controller interface
software module to generate control signals including user interface control
signals. The video display receives the user interface control signals and, in
response to the signals, interactively prompts a user to provide user inputs to the
controller interface computer. The user inputs include a designation of a selected
group of devices from the plurality of electrically powered devices. The controller
includes circuitry adapted to provide power to the selected group of devices and to
selectively provide communication links between the electrically powered devices,
such as a communication link between one of the printer ports and the printer, as
determined by the control signals. A preferred system allows more than one user to
simultaneously select, activate and use groups of electrically powered devices. The
preferred system additionally includes mechanisms for receiving prepayment from the
users and for providing the users with receipts.

Application Filing Date (1):
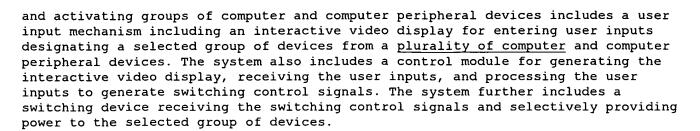19961115

Brief Summary Text (5):
Within the last few years, more and more retail outlets specializing in the rental
of computer equipment have appeared in this country. A known approach to monitoring
the usage of computers and peripheral devices is through a network such as Novell.
Such an approach to monitoring the statuses of electrically powered equipment for
the purpose of generating billing is deficient in that it is highly complicated,
requires bidirectional communications and cannot readily be adjusted to compensate
for changes in the communication interfaces and interface hardware of the monitored
devices.

Brief Summary Text (13):
In accordance with a specific illustrative embodiment of the present invention, a
system for interactively selecting and activating groups of computer and computer
peripheral devices includes a plurality of user input mechanisms adapted to receive
user inputs from a plurality of users. Each user input designates a selected group
of devices from a plurality of computer and computer peripheral devices. The system
also includes a processor adapted to receive and process the user inputs to
generate control signals. The system further includes a controller adapted to
provide power to the selected groups of devices as determined by the control
signals.

Brief Summary Text (14):
In another aspect of the present invention, a system for interactively selecting

and activating groups of computer and computer peripheral devices includes a user input mechanism including an interactive video display for entering user inputs designating a selected group of devices from a <u>plurality of computer</u> and computer peripheral devices. The system also includes a control module for generating the interactive video display, receiving the user inputs, and processing the user inputs to generate switching control signals. The system further includes a switching device receiving the switching control signals and selectively providing power to the selected group of devices.

<u>Brief Summary Text</u> (15):
In a further aspect of the present invention, a system for interactively selecting and activating groups of computer and computer peripheral devices includes a user display for providing interactive prompts. The display prompts a user to provide a user input designating a selected group of devices from a <u>plurality of computer</u> and computer peripheral devices including a printer and a <u>plurality of printers, each of the computers</u> including a printer port. The system also includes a controller with circuitry for providing power to the selected group of devices in response to the user input and for selectively providing communication links between one of the printer ports and the printer.
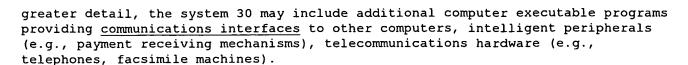
<u>Brief Summary Text</u> (19):
In another aspect of the present invention, a system for interactively selecting and activating groups of computer and computer peripheral devices includes a <u>plurality of computer</u> and computer peripheral devices, a <u>display and a controller</u>. The devices include a printer and a <u>plurality of computers,</u> each of the computers including a printer port, the <u>plurality of computers</u> including a controller interface computer which executes a controller interface software module to generate control signals including user interface control signals. The <u>display is responsive to the user interface</u> control signals and is adapted to interactively prompt a user to provide user inputs to the controller interface computer. The user inputs include a designation of a selected group of devices from the <u>plurality of computer</u> and computer peripheral devices. The controller includes circuitry adapted to provide power to the selected group of devices and to selectively provide communication links between one of the printer ports and the printer as determined by the control signals.

<u>Detailed Description Text</u> (2):
FIG. 1 shows an exemplary preferred system 30 for interactively selecting and activating groups of electrically powered devices. The system 30 includes a <u>plurality of computer</u> and computer peripheral devices such as a computer 40 and a printer 42. While the exemplary preferred system 30 includes computers, computer peripherals and other devices typically found in an office, it is additionally contemplated that the groups of electrically powered devices could include home appliances such a television or coffee maker.

<u>Detailed Description Text</u> (4):
A first key aspect of the present invention is that the system 30 provides its users with a self-service means for selecting and activating particular groups of electrical devices from a <u>plurality of electrical devices</u>. One or several controllers and a switching mechanism are required to implement the system 30. In the exemplary embodiment shown in FIG. 1, the display 44 presents a sequence of menus which interactively prompt a user of the system 30 to provide user inputs which, for example, designate a selected group of devices from a plurality of displayed computer and computer peripheral devices. At least one of the aforementioned controllers executes a computer executable program or programs for providing user interface control signals to the display 44. The switching mechanism also receives switching control signals generated by execution of a computer executable program by at least one of the controllers. In a preferred embodiment, a computer executable program is also employed to control a means for providing communication links between printer ports and a printer. As discussed below in

greater detail, the system 30 may include additional computer executable programs providing <u>communications interfaces</u> to other computers, intelligent peripherals (e.g., payment receiving mechanisms), telecommunications hardware (e.g., telephones, facsimile machines).

<u>Detailed Description Text</u> (5):
Another key aspect of the present invention is that the system 30 also includes interconnection hardware and the communications interfaces necessary for a user to "dock" a personal computer 48 or other electrically powered device to the system 30. As discussed below in greater detail, a controller or processor of the system 30 controls a sequence of interactive visual menus presented at the display 44 thereby allowing the user to select and activate a group of electrically powered devices. More specifically, the user selects a group of "stations" (to which electrically powered devices may be, but are not necessarily, connected). In a preferred embodiment, a "station" is defined as a power outlet that the system 30 is capable of selecting and providing power to, combined with a <u>communications interface</u> that is controlled and accessible by a controller or processor of the system 30. For example, a user may "dock" a notebook style personal computer 48 to the system 30 by selecting the appropriate station at the display 44. The system 30 executes software which applies power to the power outlets of the selected stations and enables the <u>communications interface</u> at the selected stations. Preferably, the <u>communications interface</u> hardware at the stations comprises connectors facilitating operable connection to currently favored computers and computer peripheral devices.
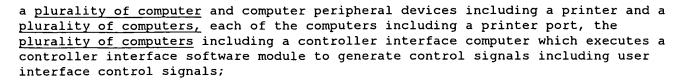
<u>Detailed Description Text</u> (8):
FIG. 2 is a block diagram showing the functional relationship between a central controller 50 and a plurality of stations adapted to receive electrically powered devices. By way of example, the central controller 50 is functionally interconnected to a station 52 and is adapted to provide power and a communication link to a facsimile machine. As shown in FIG. 2, the central controller 50 provides power and <u>communications interfaces</u> to a station 54 which, for example, is adapted to receive a controller interface computer. A station 56 is similarly controlled by the controller 50 which determines when power is made available to a copier and provides a communication link to the copier. A station 58 can, for example, provide power and communications links to an electrically interconnected computer and printer which have been configured to be used together. A station 60 is similarly configured to provide power and communications links to a modem. Additional stations such as stations 62, 64 can also be included in the system 30. The stations 62, 64 allow other computer and computer peripheral devices to be docked to and used with the system 30.

<u>Detailed Description Text</u> (10):
FIG. 9 is a functional block diagram of an alternative embodiment of the system 30. In the illustrated preferred embodiment, the controller 50 provides user interface control signals to a touch screen display. The system 30 also includes a switch box 68 which receives switching control signals from the controller 50 along control signal path 70. As shown in FIG. 9, the system 30 also includes a connection link switch box 72 which, for example, selectively provides communication links between one of the printer ports (of the controller 50 and the peripheral computer at station 58) and the peripheral printer at station 58. The operational aspects of the switch boxes 68, 72 are discussed below in greater detail. It should also be appreciated that the two switch boxes 68, 72 can be configured within a single switch box. As discussed above, the controller 50 can also be adapted to support various telephonic data <u>communication interfaces</u> such as a charge-a-call phone line interface 74, a coinless payphone equivalent.

CLAIMS:

1. A system comprising:

a plurality of computer and computer peripheral devices including a printer and a plurality of computers, each of the computers including a printer port, the plurality of computers including a controller interface computer which executes a controller interface software module to generate control signals including user interface control signals;

a display responsive to the user interface control signals and adapted to interactively prompt a user to provide user inputs to the controller interface computer, the user inputs including (i) a designation of a selected group of devices from the plurality of computer and computer peripheral devices and (ii) a prepayment method designation; and

switching control circuitry directly connected to each of the plurality of computer and computer peripheral devices through corresponding dedicated uni-directional communication lines, the switching control circuitry to control usage through deactivation of the selected group of devices and to selectively provide direct communication links between one of the printer ports and the printer as determined by the control signals.

2. A system comprising:

a plurality of computer and computer peripheral devices including a printer and a plurality of computers each of the computers including a printer port, the plurality of computers including a controller interface computer which executes a controller interface software module to generate control signals including user interface control signals;
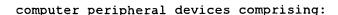
a display responsive to the user interface control signals and adapted to interactively prompt a user to provide user inputs to the controller interface computer, the user inputs including (i) a designation of a selected group of devices from the plurality of computer and computer peripheral device;

switching control circuitry directly connected to each of the plurality of computer and computer peripheral devices through corresponding dedicated uni-directional communication lines, the switching control circuitry to control usage through deactivation of the selected group of devices and to selectively provide direct communication links between one of the printer ports and the printer as determined by the control signals: and

means responsive to the control signals for receiving a prepayment from the user.

3. A system comprising:

a plurality of computer and computer peripheral devices;

a plurality of user input mechanisms adapted to receive user inputs from a plurality of users, each user input designating at least one of the plurality of computer and computer peripheral devices;

a processor adapted to receive and process the user inputs to generate control signals in response to receipt of an approval of a financial transaction initiated by one of the user inputs; and

a controller directly connected to each of the plurality of computer and computer peripheral devices through corresponding dedicated communication lines, the controller to control usage of the at least one of the plurality of computer and computer peripheral devices as determined by the control signals.

11. A system for interactively selecting and activating groups of computer and

computer peripheral devices comprising:

a user input mechanism including an interactive video display for entering user inputs designating a selected group of devices from a <u>plurality of computer</u> and computer peripheral devices;

a control module for generating the interactive video display, receiving the user inputs, and processing the user inputs by establishing communication with a remotely located device to request approval of a financial transaction and generating switching control signals in response to receiving approval; and

a switching device receiving the switching control signals and selectively providing power to the selected group of devices through corresponding dedicated communication lines.

12. A system for interactively selecting and activating groups of computer and computer peripheral devices comprising:

a display for interactively prompting a user to provide a user input designating a selected group of devices from a <u>plurality of computer</u> and computer peripheral devices including a printer and a <u>plurality of computers,</u> each of the computers including a printer port; and

a controller including circuitry for processing the user input by establishing communication with a remotely located device to request approval of a financial transaction for controlling usage of the selected group of devices and for selectively providing communication links between one of the printer ports and the printer in response to receiving approval.

☐ ▐ Generate Collection ▌

L24: Entry 6 of 31   File: USPT   May 11, 1999

DOCUMENT-IDENTIFIER: US 5903455 A
TITLE: Interface controls for use in a field device management system

Application Filing Date (1):
19961212

Brief Summary Text (22):
This invention is related to interface controls for use in a management system
capable of being coupled to one or more smart field devices. The interface controls
perform consistent communication and interfacing functions between an application,
a user interface and multiple field devices coupled to the system so that no new
programming is necessary to communicate with and display information pertaining to
newly added smart devices. The interface controls may use a communication network
which relies on a hierarchy of information related to one or more DDL's associated
with one or more smart devices connected to the system. The communication network
uses this hierarchy to call, access information from, and communicate with a DDS
associated with the one or more categorized DD's, smart devices connected within a
system, and/or a database associated with the system.

Detailed Description Text (22):
Likewise, the DCI 60 recognizes or receives changes in OLE objects stored within
the memory associated with the server network 66 and performs functions based
thereon to implement communication with the current application 56 and the
interface block 58. The device server 68 is essentially a set of software routines
which have a specified correspondence with the set of OLE objects in the determined
OLE hierarchy. These routines are specifically developed to communicate with a DDS
72, a smart device communication interface 74, and the OLE objects of the defined
hierarchy. Such routines may, for example, transmit, retrieve, and change
particular types of data and information stored within, or available from, the
smart devices within the process 12 and/or from DD's (which are files) associated
with the smart devices within the process 12. Likewise, the database server 70 is
essentially a set of software routines associated with the OLE objects in the
determined OLE hierarchy. These routines communicate with the DDS or API 72 and/or
an FMS database interface 8CI to, for example, transmit, retrieve, or change
particular types of data and information stored within, or available from, the FMS
database 40 and/or from the DD's which are associated with the smart devices for
which data is stored in the FMS database 40. As indicated in FIG. 2, the DD's used
by the DDS 72 are stored in a device description library 76 coupled to the DDS
library 72.

Detailed Description Text (26):
If the specific data was available from the DD, the server 68 stores and maintains
that data in the OLE object to which the retrieved data is related. If however, the
requested specific data is not available from the DD for a device or a block of a
device but is stored, instead, in the on-line device, the server 68 sends a command
to the smart device communication interface 74 (which may comprise any known smart
device communication interface including, for example, a Fieldbus device interface
developed by SoftIng, a German company located in Karlsruhe, or the HART device
interface of Micromotion, located in Boulder, Colo.) to retrieve the specific data
from the on-line device.

Detailed Description Text (27):
The smart device communication interface 74 then sends a request to the DDS 72 for information on how to get the specific on-line device for the data requested by the server 68. The DDS 72 retrieves this instruction information from the DD for the on-line device and returns the instruction information to the smart device communication interface 74 which, in turn, sends a proper request to the on-line smart device. The smart device then responds with a data stream including the specific data. The smart device communication interface 74 then sends a request to the DDS 72 for information on how to interpret the data stream received from the on-line smart device. The DDS 72 then retrieves interpretation instructions from the DD for the on-line smart device and returns them to the smart device communication interface 74 which, in turn, interprets the data stream from the on-line device in accordance with the interpretation instructions in order to extract the specific data requested by the server 68. The smart device communication interface then returns the specific data to the server 68 which provides the retrieved data to the OLE object with which that data is associated.

Detailed Description Text (28):
The process of writing data to an on-line device is similar to the process of reading data from that device except that the server 68 first sends a request to the DDS 72 for write information, e.g., whether the data is writable, what type, specific values and range of data can be written, etc. If the data is writable, the server 68 sends a write command to the smart device communication interface 74 which, in turn, interfaces with the DDS 72 for write protocols for the on-line device and sends the proper write command to the on-line device in response to the information. The smart device communication interface 74 can also interpret other data from the on-line devices, such as write verifications, response codes, data or value changes which occur in the device, etc. and sends such data to the server 68 for storage in the proper OLE object.
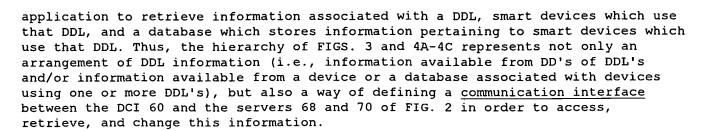
Detailed Description Text (29):
In some instances, the DDS 72 will inform the server 68 that it needs more information to answer a request for data. For example, the DDS 72 may determine that the handling property of a parameter (i.e., whether the parameter is readable and/or writable) is dependent on the mode parameter of a particular device. The DDS 72 sends a request to the server 68 for the mode parameter of the device. In response thereto, the server 68 sends a request for the mode parameter of a device to the smart device communication interface 74 which operates as described above to retrieve the mode parameter of the device. When the server 68 receives the mode parameter of the device from the smart device communication interface 74, it sends this information to the DDS 72 which, thereafter, determines the handling property of a parameter of a device and returns such property to the server 68 which, in turn, places that value in the proper OLE parameter object.

Detailed Description Text (30):
Communication between the server 70, the DDS 72 and the FMS database interface 80 is similar to that described above, except that the FMS database interface 80 is programmed to read and write information to and from the FMS database 40 instead of a smart device. Generally, however, the FMS database interface 80 mimics the functions of the smart device communication interface 74 as they relate to communications between the DDS 72 and the server 70.

Detailed Description Text (34):
FIGS. 3 and 4A-4C illustrate a particular hierarchy of OLE objects which has been developed to represent all of the information defined within or available from one or more DDL's, a set of smart devices which follow the protocols of those DDL's and a database which stores information related to devices using those DDL's. The hierarchy of FIGS. 3 and 4A-4C also represents the relationships between those OLE objects. This hierarchy can be used within an OLE environment to enable an

application to retrieve information associated with a DDL, smart devices which use that DDL, and a database which stores information pertaining to smart devices which use that DDL. Thus, the hierarchy of FIGS. 3 and 4A-4C represents not only an arrangement of DDL information (i.e., information available from DD's of DDL's and/or information available from a device or a database associated with devices using one or more DDL's), but also a way of defining a communication interface between the DCI 60 and the servers 68 and 70 of FIG. 2 in order to access, retrieve, and change this information.

Detailed Description Text (94):
FIG. 12 illustrates the operation of a control in response to a message from the user interface. A block 370 checks to determine if the user action is meaningful. The block 370 may, for example, determine if the user clicked the proper button of the mouse or if the pointer (i.e., the cursor or arrow) was located within an area of the control display where the control :recognizes the user's actions as requests for action. If the user action is not meaningful, a block 372 simply ignores the user action or gives some indication that the action has been ignored (e.g., refreshing the user display with the same display interface attributes). Thereafter, control is returned to the block 315.

CLAIMS:

25. A control adapted for use by a management system that includes a user interface and that is capable of being coupled to a plurality of field devices, each having a device description associated therewith, the control comprising:

means for communicating with the plurality of field devices and the device descriptions to effect communication with respect to a multiplicity of groups of logically related items of device data stored in the plurality of field devices and the device descriptions;

means for implementing a particular control for a particular one of the multiplicity of groups of logically related items of device data including,

means for controlling the communicating means to retrieve the items of device data within the particular one of the multiplicity of groups of logically related items of device data to produce a retrieved group of items of device data, and

means for displaying the retrieved group of items of device data via the user interface in a predefined format;

means for identifying any of the multiplicity of groups of logically related items of device data as the particular one of the multiplicity of groups of logically related items of device data; and

means responsive to the identifying means for invoking the implementing means to create the particular control for the particular one of the multiplicity of groups of logically related items of device data.

33. The control of claim 31, wherein the communicating means includes means for categorizing the items of device data into a predetermined hierarchy of categories of device data, each having communication instructions, and wherein the communicating means communicates with one of the plurality of devices, one of the device descriptions or the database using the communication instructions associated with one of the categories of the hierarchy of categories of device data.

35. The control of claim 31, wherein the retrieved group of items of device data comprises data pertaining to one of the plurality of field devices, and wherein a first item of the retrieved group of items of device data includes a pictorial representation of the one of the plurality of field devices.

●   ●

36. The control of claim 35, wherein a second item of the retrieved group of items of device data includes an indication of the type of the one of the plurality of field devices.

37. The control of claim 31, wherein the retrieved group of items of device data comprises data pertaining to a block comprising one of an input, an output, or a control function associated with one of the plurality of field devices, and wherein a first item of the retrieved group of items of device data includes a pictorial representation of the block.

38. The control of claim 31, wherein the retrieved group of items of device data comprises data pertaining to a parameter associated with one of the plurality of field devices, and wherein a first item of the retrieved group of items of device data includes a label of the parameter and a second item of the retrieved group of items of device data includes a value of the parameter.

40. The control of claim 31, wherein the retrieved group of items of device data comprises configuration data related to the configuration of one of the plurality of field devices at a particular time, and wherein. the implementing means includes means for allowing a user to change the particular time associated with the configuration data.

42. The control of claim 31, wherein the retrieved group of items of device data comprises configuration data related to first and second configurations associated with one or two of the plurality of field devices at two individual times, and wherein the implementing means includes means for allowing a user to change the individual time associated with the configuration data related to the first configuration.

44. A method of controlling communication between a management system, having a user interface, and a plurality of field devices, each having a device description associated therewith, the method comprising the steps of:

providing a generalized interfacing routine which, for any particular one of a plurality of different groups of logically related items of the device data performs the steps of;

retrieving the items of device data associated with the particular one of the groups of logically related items of device data as stored in the plurality of field devices and the device descriptions,

displaying the retrieved items of device data associated with the particular one of the groups via the user interface in a predefined format,

recognizing when one of the items of device data stored in one of the plurality of field devices changes, and

indicating, via the user interface, the change to the one of the items of device data when the one of the items of device data corresponds to one of the retrieved items of device data associated with the particular one of the groups; and

using the generalized interfacing routine to perform communication for a multiplicity of the plurality of different groups of logically related items of device data.